


DATA ACCESS CONTROLLER AND RECORD MEDIUM RECORDED WITH DATA ACCESS CONTROL PROGRAM

Patent number: JP11119923
Publication date: 1999-04-30
Inventor: INAMOTO YASUSHI
Applicant: FUJITSU LTD
Classification:
- **International:** G06F3/06; G06F3/06
- **European:**
Application number: JP19970287372 19971020
Priority number(s):

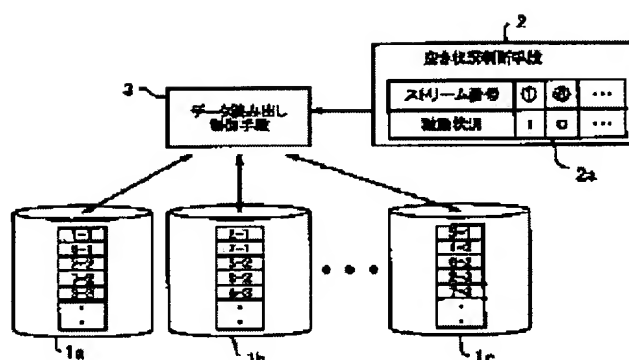
Also published as:

 US6049891 (A1)

Abstract of JP11119923

PROBLEM TO BE SOLVED: To decrease data lost due to read errors, when reading data while guaranteeing real time property.

SOLUTION: Storage devices 1a-1c store a series of data, while distributing them by each block. An idle condition discriminating means 2 investigates an idle time zone, in which the read processing of data from the plural storage devices 1a-1c is not performed. A data read control means 3 reads preceding read data to be read first after the idle time band, while utilizing the idle time band. When the read error occurs during that read, read retry is performed while utilizing idle time, until data output time assigned to the preceding read data. In this case, the read retry is performed repeatedly, as many number of times of read retries as are executable within allowed time as an upper limit.



Data supplied from the esp@cenet database - Worldwide

(19) 日本国特許庁 (J P)

(12) 公開特許公報 (A)

(11) 特許出願公開番号

特開平11-119923

(43) 公開日 平成11年(1999) 4月30日

(51) Int.Cl.⁵
G 0 6 F 3/06

識別記号
5 4 0
3 0 5

F I
G 0 6 F 3/06

5 4 0
3 0 5 K

審査請求 未請求 請求項の数10 O L (全 26 頁)

(21) 出願番号 特願平9-287372

(22) 出願日 平成9年(1997)10月20日

(71) 出願人 000005223

富士通株式会社

神奈川県川崎市中原区上小田中4丁目1番
1号

(72) 発明者 稲本 康

神奈川県川崎市中原区上小田中4丁目1番
1号 富士通株式会社内

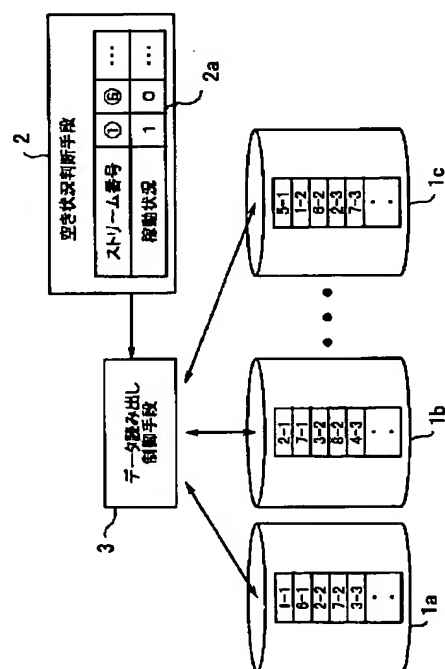
(74) 代理人 弁理士 服部 毅巖

(54) 【発明の名称】 データアクセス制御装置及びデータアクセス制御プログラムを記録した媒体

(57) 【要約】

【課題】 リアルタイム性を保証したデータの読み出しを行う際に、リードエラーによって失われるデータを減少させる。

【解決手段】 記憶装置1a～1cには、一連のデータが1ブロック毎に分散して格納されている。空き状況判断手段2は、複数の記憶装置1a～1cからのデータの読み出し処理が行われていない空き時間帯を調べる。データ読み出し制御手段3は、空き時間帯の後に最初に読み出すべき先行リードデータを、空き時間帯を利用して記憶装置から読み出す。その読み出しの間にリードエラーが発生した場合には、先行リードデータに割り当てられているデータ出力時刻までの空き時間を利用してリードリトライを行う。この際、許容時間内で実行可能なリードリトライの回数を上限として、リードリトライを繰り返す。



【特許請求の範囲】

【請求項1】 1つの伝送路に接続された複数の記憶装置に1ブロック毎に分散して格納されている一連のデータを、前記複数の記憶装置内部で順次読み出し処理を行い、読み出された1ブロックのデータを、予め定められたデータ出力時間帯内において前記伝送路を介して転送するデータアクセス制御装置において、前記複数の記憶装置でのデータの読み出し処理が行われない空き時間帯を調べる空き状況判断手段と、前記空き時間帯の後に最初に読み出すべきデータを先行リードデータとし、前記空き時間帯を利用して前記先行リードデータの読み出し処理を行い、リードエラーが発生した場合には、前記先行リードデータに割り当てられている前記データ出力時間帯までの時間を利用してリードリトライを行うデータ読み出し制御手段と、を有することを特徴とするデータアクセス制御装置。

【請求項2】 前記空き状況判断手段は、1ブロックのデータを前記記憶装置から読み出して前記伝送路へ出力するのに必要な時間間隔で分割した時間帯をタイムスロットとし、連続して読み出し処理が行われていないタイムスロットの数にタイムスロットの時間を乗じたものを許容時間とし、前記データ読み出し制御手段は、前記許容時間内で実行可能なリードリトライ回数を上限として、リードリトライを繰り返すことを特徴とする請求項1記載のデータアクセス制御装置。

【請求項3】 前記空き状況判断手段は、一連のデータへのアクセス環境の割当をストリームとし、同時に読み出し可能な最大数のストリームのそれぞれに前記記憶装置毎のタイムスロットを割り当て、与えられた時刻に達したタイムスロットが割り当てられている対応ストリームが稼働中でない場合には、前記対応ストリームに割り当てられている全てのタイムスロットが空いていると判断することを特徴とする請求項2記載のデータアクセス制御装置。

【請求項4】 前記空き状況判断手段は、ストリームの番号とそれぞれのストリームが稼働中か否かの情報との組を、割り当てられたタイムスロットの出現順に並べたストリーム稼働状況テーブルを用意し、判断対象となるストリームが稼働中でない場合には、稼働中と設定されているストリームが見つかるまで前記ストリーム稼働状況テーブルを後方に向かって検索し、見つかるまでに進んだ欄の数を、連続して読み出し処理が行われていないタイムスロットの数とすることを特徴とする請求項3記載のデータアクセス制御装置。

【請求項5】 1つの伝送路に接続された複数の記憶装置に1ブロック毎に分散して格納すべき一連のデータを、予め定められたデータ入力時間帯内において前記伝送路を介して転送し、前記複数の記憶装置へ順次書き込むデータアクセス制御装置において、

前記複数の記憶装置へのデータの書き込み処理が行われない空き時間帯を調べる空き状況判断手段と、前記空き時間帯の直前に書き込むべきデータの書き込みの際にライトエラーが発生した場合には、前記空き時間帯を利用してライトリトライを行うデータ書き込み制御手段と、を有することを特徴とするデータアクセス制御装置。

【請求項6】 前記空き状況判断手段は、1ブロックのデータを前記記憶装置に書き込むのに必要な時間間隔で分割した時間帯をタイムスロットとし、連続して書き込み処理が行われていないタイムスロットの数にタイムスロットの時間を乗じたものを許容時間とし、前記データ書き込み制御手段は、前記許容時間内で実行可能なライトリトライ回数を上限として、ライトリトライを繰り返すことを特徴とする請求項5記載のデータアクセス制御装置。

【請求項7】 前記空き状況判断手段は、一連のデータへのアクセス環境の割当をストリームとし、同時に読み出し可能な最大数のストリームのそれぞれに前記記憶装置毎のタイムスロットを割り当て、与えられた時刻に達したタイムスロットが割り当てられている対応ストリームが稼働中でない場合には、前記対応ストリームに割り当てられている全てのタイムスロットが空いていると判断することを特徴とする請求項6記載のデータアクセス制御装置。

【請求項8】 前記空き状況判断手段は、前記ストリームの番号と、それぞれが稼働中か否かの情報との組を、割り当てられたタイムスロットの出現順に並べたストリーム稼働状況テーブルを用意し、判断対象となるストリームが稼働中でない場合には、稼働中と設定されているストリームが見つかるまで前記ストリーム稼働状況テーブルを後方に向かって検索し、見つかるまでに進んだ欄の数を、連続して読み出し処理が行われていないタイムスロットの数とすることを特徴とする請求項7記載のデータアクセス制御装置。

【請求項9】 1つの伝送路に接続された複数の記憶装置に1ブロック毎に分散して格納されている一連のデータを、前記複数の記憶装置内部で順次読み出し処理を行い、読み出された1ブロックのデータを、予め定められたデータ出力時間帯内において前記伝送路を介して転送するためのデータアクセス制御プログラムを記録した媒体において、

前記複数の記憶装置でのデータの読み出し処理が行われない空き時間帯を調べる空き状況判断手段、前記空き時間帯の後に最初に読み出すべきデータを先行リードデータとし、前記空き時間帯を利用して前記先行リードデータの読み出し処理を行い、リードエラーが発生した場合には、前記先行リードデータに割り当てられている前記データ出力時間帯までの時間を利用してリードリトライを行うデータ読み出し制御手段、

としてコンピュータを機能させることを特徴とするデータアクセス制御プログラムを記録した媒体。

【請求項10】 1つの伝送路に接続された複数の記憶装置に1ブロック毎に分散して格納すべき一連のデータを、予め定められたデータ入力時間帯内において前記伝送路を介して転送し、前記複数の記憶装置へ順次書き込むためのデータアクセス制御プログラムを記録した媒体において、

前記複数の記憶装置へのデータの書き込み処理が行われない空き時間帯を調べる空き状況判断手段、

前記空き時間帯の直前に書き込むべきデータの書き込みの際にライトエラーが発生した場合には、前記空き時間帯を利用してライトリトライを行うデータ書き込み制御手段、

としてコンピュータを機能させることを特徴とするデータアクセス制御プログラムを記録した媒体。

【発明の詳細な説明】

【0001】

【発明の属する技術分野】本発明は磁気ディスクなどからのデータの入出力を行うデータアクセス制御装置及びデータアクセス制御プログラムを記録した媒体に関し、特にリアルタイム性を保証しつつデータの入出力を行うデータアクセス制御装置及びデータアクセス制御プログラムを記録した媒体に関する。

【0002】

【従来の技術】ビデオ・オン・デマンドに使用するビデオサーバは、大量のデータを磁気ディスク装置（以下、単に「ディスク」と呼ぶ）から読み出しながら、リアルタイム性を保証しつつ、ネットワークを介して接続された複数のクライアントへ送出しなければならない。このため、ディスクからデータを読み出す方法について特別な工夫が必要となる。

【0003】例えば、6.144Mb/s（メガビット/秒）のMPEG（Motion Picture Experts Group）-2のビデオストリームを同時に14本出力する場合、合計データ出力速度は約86Mb/s＝約10.8MB/s（メガバイト/秒）になる。しかし、現在最高性能のディスクを使っても、単体では長時間の平均出力速度がせいぜい4～5MB/sしか出せない。しかも、複数ストリームのデータを均等に読み出すためには、データを小さいブロックに分割して、各ストリームのブロックを順番にアクセスする必要があるが、そのたびにシーク動作と回転待ち動作を行うため、平均速度はさらに低くならざるを得ない。

【0004】そこで、ビデオサーバでは、ディスクを単体ではなく複数台使用し、すべてのストリームのデータをブロックごとに分割して複数のディスクに所定の順番

$$b = (384 / 20) \text{ ms} + (\text{オーバーヘッド}) \approx 35 \text{ ms}$$

・・・・・・（2）

上記のオーバーヘッドとは、ディスクインタフェースから

で格納する、いわゆるストライピングを行う。ストライピングを行った場合、複数のディスクは1本のデータバスで接続される。そして、各ディスクから読み出されたデータは、ディスクインタフェースを介してバッファメモリに一旦蓄えられ、各ストリームごとに速度を一定に保つように制御されつつ、ネットワークインタフェースを介してネットワークへ出力され、クライアントへ配信される。

【0005】図12は、データの読み出し状況を示す図である。これは、5台のディスクがデータにアクセスしてそれをデータバスに出力する様子を表している。この図において、薄く塗りつぶした部分は、ディスクのシーク動作、回転待ち動作、及び記憶媒体からディスク装置内のキャッシュメモリへのデータ読み出し動作（以後特に断りのない限り、これらの3つの動作を合わせて「データの読み出し」と呼ぶ）の合計時間aを表している。網掛けの部分は読み出されたデータがディスク装置内のキャッシュメモリからデータバスへ出力される時間bを表す。

【0006】各ディスクはデータバスを共用しているため、同時に複数のディスクがデータを出力することはできない。そこで、データバスへの出力のタイミングを各ディスクでずらすことにより、データバスの利用効率を向上させている。すなわち、あるディスクがデータバスへ出力している間、他のディスクはシーク動作や回転待ち動作やディスク読み出し動作を行う。この方法により、長時間平均出力速度に関して、単体ディスクの場合よりも高い性能を出すことが可能になる。

【0007】一例として、長時間平均出力速度の上限（Sustained Data Transfer Rate）が4MB/s、最大シーク時間が17ms、最大回転待ち時間が8.3msのディスクを使う場合を取り上げる。6.144Mb/sのMPEG-2データをディスクにストライピングして格納する際のブロックサイズを0.5秒分すなわち6.144Mb×0.5＝3.073Mb＝384KBとする。データバスをWide Fast SCSI（SCSI（Small Computer System Interface）を高速化し、更にバス幅を16ビットに拡張した拡張規格）とすれば、その転送速度は20MB/sであるから、図12のaおよびbの時間は次のようになる。

【0008】

【数1】

$$a = 17 \text{ ms} + 8.3 \text{ ms} + (384 / 4) \text{ ms} + (\text{オーバーヘッド}) \approx 140 \text{ ms} \cdots \cdots (1)$$

【0009】

【数2】

ディスクへのコマンドやディスクからディスクインタフ

エースへの応答がデータバス上を流れる時間を指す。この例ではディスク1台あたり

$$384 \text{ KB} / (140 \text{ ms} + 35 \text{ ms}) = \text{約} 2.2 \text{ MB/s} \\ \dots\dots (3)$$

の長時間平均出力速度が得られるため、ディスク5台の合計では

【0011】

【数4】

$$2.2 \text{ MB/s} \times 5 = 11.0 \text{ MB/s} \\ 11.0 \times 8 / 6.144 = 14.3 \dots\dots (4)$$

となって、6.144 Mb/sのストリーム14本を同時に出力できることになる。

【0012】

【発明が解決しようとする課題】しかし、ディスクでは一般的にリードエラーが避けられないため、以下のような問題点がある。

【0013】通常のシステムではディスクにリードエラーが発生した場合、正しく読めるまで上限回数を定めてリードリトライを行う。正しく読めた時点で正常終了とし、上限回数までリトライしても正しく読めない場合はエラーとして処理を終了する。リードリトライを1回行うには、通常、記憶媒体が1回転する時間が余計にかかる（先の例にあげたディスクでは最大回転待ち時間と同じ8.3ms）。

【0014】従来のビデオサーバでは、最大性能発揮時（最多ストリーム送出時）のリアルタイム性を保証するために、タイムスロットの考え方に基づいて設計されて

・M=1のとき

$$384 / (140 + 35 + 8.3 \times 1) \times 5 = \text{約} 10.5 \text{ MB/s} \\ 10.5 \times 8 / 6.144 = 13.6 \rightarrow 13 \text{ ストリーム} \\ \dots\dots (5)$$

【0018】

・M=2のとき

$$384 / (140 + 35 + 8.3 \times 2) \times 5 = \text{約} 10.0 \text{ MB/s} \\ 10.0 \times 8 / 6.144 = 13.0 \rightarrow 13 \text{ ストリーム} \\ \dots\dots (6)$$

【0019】

・M=3のとき

$$384 / (140 + 35 + 8.3 \times 3) \times 5 = \text{約} 9.6 \text{ MB/s} \\ 9.6 \times 8 / 6.144 = 12.5 \rightarrow 12 \text{ ストリーム} \\ \dots\dots (7)$$

というように、6.144 Mb/sのストリームを14本出すことが不可能になる。

【0020】一方、現実のビデオ・オン・デマンドのビデオサーバでは、最多ストリーム数を常時出力し続ける必要はまずない。クライアントからの要求が少ないときは、ビデオサーバから出力するストリーム数も少なくなる。しかし、従来のビデオサーバでは、クライアントからの要求の多少にかかわらず、1タイムスロットの時間内に1ブロックの読み出しとデータバスへの出力を終え

【0010】

【数3】

いる。すなわち、図12のa+bの時間を1タイムスロットとし、この時間内に1ブロックの読み出しとデータバスへの出力を終えるように設計されている。

【0015】ストライピングにより、図12のタイムチャートのように各ディスクが目一杯の性能で稼働すれば、全体として最大性能を発揮できる。しかし、この場合は各ディスクに時間的な余裕がなく、仮にリードエラーが発生してもリードリトライを行うことができない。したがって、リードエラーが発生すれば、即座にエラー終了となるが、そのときはデータが欠落する。これにより、例えばクライアントの画面が0.5秒間ブラックアウトするような事態が起こる。

【0016】これを避ける一つの方法は、あらかじめリードリトライを行う時間の余裕を持たせて、タイムスロットの時間を設定することである。すなわち、リードエラーが発生したときリトライをM回まで行うようにするには、図12のa+bにディスクの記憶媒体がM回転する時間（先の例では8.3ms×M）を加算した値を1タイムスロットとすればよい。しかしながら、この方法ではリードリトライを全く行わない場合に比べて最大性能（最多同時ストリーム数）が低下する。先の例では、

【0017】

【数5】

【数6】

【数7】

るよう設計されている。このため、ビデオサーバのからの出力ストリーム数が少なく、各ディスクに時間的余裕がある場合でも、リードエラーに対するリトライが行われることはなかった。

【0021】なお、以上の問題はディスクからデータを読み出す場合に限ったものではなく、リアルタイム性を保証したデータを複数のディスクへストライピングにより書き込む場合も同様である。すなわち、1タイムスロットの時間内にデータバスからの入力と記憶媒体への書

き込みとを終えるように設計されているため、最多ストリーム数で書き込みを行っていない場合においても、ライトエラーに対するライトリトライを行うことができなかった。

【0022】本発明はこのような点に鑑みてなされたものであり、リアルタイム性を保証したデータの読み出しを行う際に、リードエラーによって失われるデータを減少させることができるデータアクセス制御装置を提供することを目的とする。

【0023】また、本発明の他の目的は、リアルタイム性を保証したデータの書き込みを行う際に、ライトエラーによって失われるデータを減少させることができるデータアクセス制御装置を提供することである。

【0024】また、本発明の他の目的は、リアルタイム性を保証したデータの読み出しを行う際に、リードエラーによって失われるデータを減少させることができるデータアクセス制御プログラムを記録した媒体を提供することである。

【0025】また、本発明の他の目的は、リアルタイム性を保証したデータの書き込みを行う際に、ライトエラーによって失われるデータを減少させることができるデータアクセス制御プログラムを記録した媒体を提供することである。

【0026】

【課題を解決するための手段】本発明では上記課題を解決するために、1つの伝送路に接続された複数の記憶装置に1ブロック毎に分散して格納されている一連のデータを、前記複数の記憶装置内部で順次読み出し処理を行い、読み出された1ブロックのデータを、予め定められたデータ出力時間帯内において前記伝送路を介して転送するデータアクセス制御装置において、前記複数の記憶装置でのデータの読み出し処理が行われない空き時間帯を調べる空き状況判断手段と、前記空き時間帯の後に最初に読み出すべきデータを先行リードデータとし、前記空き時間帯を利用して前記先行リードデータの読み出し処理を行い、リードエラーが発生した場合には、前記先行リードデータに割り当てられている前記データ出力時間帯までの時間を利用してリードリトライを行うデータ読み出し制御手段と、を有することを特徴とするデータアクセス制御装置が提供される。

【0027】このデータアクセス制御装置によれば、空き状況判断手段により、複数の記憶装置でのデータの読み出し処理が行われない空き時間帯が調べられる。そして、データ読み出し制御手段により、空き時間帯の後に最初に読み出すべきデータを先行リードデータとされ、空き時間帯を利用して先行リードデータの読み出し処理が行われ、リードエラーが発生した場合には、先行リードデータに割り当てられているデータ出力時間帯までの時間を利用してリードリトライが行われる。

【0028】また、1つの伝送路に接続された複数の記

憶装置に1ブロック毎に分散して格納すべき一連のデータを、予め定められたデータ入力時間帯内において前記伝送路を介して転送し、前記複数の記憶装置へ順次書き込むデータアクセス制御装置において、前記複数の記憶装置へのデータの書き込み処理が行われない空き時間帯を調べる空き状況判断手段と、前記空き時間帯の直前に書き込むべきデータの書き込みの際にライトエラーが発生した場合には、前記空き時間帯を利用してライトリトライを行うデータ書き込み制御手段と、を有することを特徴とするデータアクセス制御装置が提供される。

【0029】このデータアクセス制御装置によれば、空き状況判断手段により、複数の記憶装置へのデータの書き込み処理が行われない空き時間帯が調べられる。そして、空き時間帯の直前に書き込むべきデータの書き込みの際にライトエラーが発生した場合には、データ書き込み制御手段により、空き時間帯を利用したライトリトライが行われる。

【0030】また、1つの伝送路に接続された複数の記憶装置に1ブロック毎に分散して格納されている一連のデータを、前記複数の記憶装置内部で順次読み出し処理を行い、読み出された1ブロックのデータを、予め定められたデータ出力時間帯内において前記伝送路を介して転送するためのデータアクセス制御プログラムを記録した媒体において、前記複数の記憶装置でのデータの読み出し処理が行われない空き時間帯を調べる空き状況判断手段、前記空き時間帯の後に最初に読み出すべきデータを先行リードデータとし、前記空き時間帯を利用して前記先行リードデータの読み出し処理を行い、リードエラーが発生した場合には、前記先行リードデータに割り当てられている前記データ出力時間帯までの時間を利用してリードリトライを行うデータ読み出し制御手段、としてコンピュータを機能させることを特徴とするデータアクセス制御プログラムを記録した媒体が提供される。

【0031】この媒体に記録されたデータアクセス制御プログラムをコンピュータで実行すれば、複数の記憶装置でのデータの読み出し処理が行われない空き時間帯を調べる空き状況判断手段と、空き時間帯の後に最初に読み出すべきデータを先行リードデータとし、空き時間帯を利用して先行リードデータの読み出し処理を行い、リードエラーが発生した場合には、先行リードデータに割り当てられているデータ出力時間帯までの時間を利用してリードリトライを行うデータ読み出し制御手段と、の機能がコンピュータで実現される。

【0032】また、1つの伝送路に接続された複数の記憶装置に1ブロック毎に分散して格納すべき一連のデータを、予め定められたデータ入力時間帯内において前記伝送路を介して転送し、前記複数の記憶装置へ順次書き込むためのデータアクセス制御プログラムを記録した媒体において、前記複数の記憶装置へのデータの書き込み処理が行われない空き時間帯を調べる空き状況判断手

段、前記空き時間帯の直前に書き込むべきデータの書き込みの際にライトエラーが発生した場合には、前記空き時間帯を利用してライトリトライを行うデータ書き込み制御手段、としてコンピュータを機能させることを特徴とするデータアクセス制御プログラムを記録した媒体が提供される。

【0033】この媒体に記録されたデータアクセス制御プログラムをコンピュータで実行すれば、複数の記憶装置へのデータの書き込み処理が行われない空き時間帯を調べる空き状況判断手段と、空き時間帯の直前に書き込むべきデータの書き込みの際にライトエラーが発生した場合には、空き時間帯を利用してライトリトライを行うデータ書き込み制御手段と、の各機能がコンピュータによって実現される。

【0034】

【発明の実施の形態】以下、本発明の実施の形態を図面を参照して説明する。図1は、本発明の原理構成図である。このデータアクセス制御装置には、1つの伝送路に接続された複数の記憶装置1a～1cが設けられている。記憶装置1a～1cには、一連のデータが1ブロック毎に分散して格納されている。すなわち、ストライピングが行われている。

【0035】空き状況判断手段2は、複数の記憶装置1a～1cにおいてデータの読み出し処理が行われない空き時間帯を調べる。具体的には、1ブロックのデータを記憶装置1a～1cから読み出すのに必要な時間間隔で分割した時間帯をタイムスロットとする。さらに、一連のデータへのアクセス環境の割当をストリームとし、同時に読み出し可能な最大数のストリームのそれぞれに予めタイムスロットを割り当てる。そして、あるタイムスロットの時刻に達すると、そのタイムスロットに割り当てられたストリームの稼働状況を見る。稼働状況の判断には、ストリーム稼働状況テーブル2aを用いる。ストリーム稼働状況テーブル2aには、ストリームの番号とそれぞれが稼働中か否かの情報との組が、タイムスロットの出現順に並べられている。判断対象となるストリームが稼働中でない場合には、稼働中と設定されているストリームが見つかるまでストリーム稼働状況テーブル2aを後方に向かって検索し、見つかるまでに進んだ欄の数を、連続した空きタイムスロットの数とする。連続した空きタイムスロットの数にタイムスロットの時間を乗じたものが許容時間となる。

【0036】データ読み出し制御手段3は、空き時間帯の後に最初に読み出すべき先行リードデータを、空き時間帯を利用して記憶装置から読み出す。その読み出しの間にリードエラーが発生した場合には、先行リードデータに割り当てられているデータ出力時刻までの空き時間を利用してリードリトライを行う。この際、許容時間内で実行可能なリードリトライの回数を上限として、リードリトライを繰り返し行う。

【0037】このように、空き時間がある場合には次に読み出すべきデータの読み出し処理を先行して行い、リードエラーがあった場合には、データ出力までの時間を利用してリードリトライを行うため、リードエラーが発生しても、その一部のデータはリードリトライによって読み出すことが可能である。

【0038】以下に、本発明をビデオサーバシステムに適用した場合を例にとり、本発明の実施の形態を具体的に説明する。図2は、ビデオサーバシステムの構成例を示す図である。これは、5台のディスク（HDD）21～25に格納されている情報を最大14人のユーザに提供するものである。即ち、最大ストリーム数は「14」である。

【0039】中央演算処理装置（CPU）11は、システムバス10を介して接続されたメモリ12、ディスクインタフェース13、バッファメモリ14及びネットワークインタフェース15を制御する。

【0040】メモリ12は、CPU11が実行するプログラムや各種データを一時的に格納すると共に、ストリーム稼働状況テーブルを格納する。ストリーム稼働状況テーブルとは、どのストリームが稼働しているのかを示すテーブルである。

【0041】図3は、ストリーム稼働状況テーブル40の例を示す図である。このように、ストリーム稼働状況テーブル40は、ストリーム番号と現在の稼働状況（1＝稼働中／0＝非稼働）を、全ストリーム稼働時にデータバスへデータを出力する順番に並べたものである。

【0042】図2に戻り、ディスクインタフェース13は、データバス20を介してアクセス要求を出力することにより、ディスク21～25との間でデータの送受信を行う。ディスクインタフェース13から各ディスクへのアクセス順は、ディスク21→ディスク22→ディスク23→ディスク24→ディスク25→ディスク21・・・の順である。データバス20は、実際にはSCSIバスやSCSI-2バスなどである。

【0043】バッファメモリ14は、ディスクインタフェース13が各ディスク21～25から受け取ったデータを一時的に格納する。ネットワークインタフェース15は、バッファメモリ14内のデータをネットワーク30を介して接続されたクライアント31、32へ配信する。

【0044】ディスク21～25には、複数のビデオデータがストライピングにより格納されている。ストライピングの際の単位データのサイズは、1つのタイムスロット内で出力できるようなサイズである。各ディスク21～25にはそれぞれ、1～5までのディスク番号が割り当てられている。ここで、ディスク番号とは、各ディスクを識別するための番号である。ここでは、1からディスク台数までの自然数である。各ディスクはディスク番号の順に動作し、最後まで行ったらディスク番号1の

ディスクへ戻る。以後、「次のディスク番号」といった場合には、動作順に従って次に動作させるべきディスク番号を指す。

【0045】図4は、ディスク上のデータ配置の例を示す図である。この例では、5台のディスク21～25に9種類のファイルをストライピングして配置している。図の中にある数字（2～3など）は何番目のファイルの何番目のブロックかを表している。ひとつのファイルに属するブロックを1台のディスクに連続して配置せず、複数のディスクへ均等に分散して配置することが肝要である。例えば1番目のファイルのデータは、「ディスク1」、「ディスク5」、「ディスク4」、「ディスク3」、「ディスク2」、「ディスク1」・・・の順で格納される。格納される順番は、他のファイルも同じである。

【0046】このような構成のビデオサーバシステムにおいて、ビデオ画像の要求がクライアントからあると、そのクライアントに対してストリームが割り当てられる。ストリームには、各ディスクにおけるタイムスロットが割り当てられており、そのタイムスロットの時間帯を利用して、クライアントから要求された一連のデータを読み出す。なお、図4から分るように、一連のデータは、「ディスク1」、「ディスク5」、「ディスク4」、「ディスク3」、「ディスク2」、「ディスク1」・・・の順で格納されているため、ストリームに対して割り当てられているタイムスロットも、同じ順番で読み出せるように配分されている。

【0047】ディスクから読み出されたデータはディスクインタフェース13を介してバッファメモリ14に一旦蓄えられ、各ストリームごとに速度を一定に保つように制御されつつ、ネットワークインタフェース15を介してネットワーク30へ出力される。そして、クライアント31、32へ配信される。

【0048】次に、データの読み出し処理の詳細を説明する。図5は、データ読み出し手順を示すフローチャートである。これは、CPU11が、メモリ12内にロードされた所定のプログラムに従って実行する処理である。

【S1】ディスク番号を示す値「d」を1に設定し、ストリーム番号を示す値「s」を1に設定する。

【S2】sで示されたストリームへの要求が、新規ストリームの送出開始又は既存ストリーム送出終了の要求であるか否かを判断する。これらのいずれかの要求であればステップS3に進み、これらの要求でなければステップS4に進む。

【S3】ストリーム稼働状況テーブルの内容を更新する。具体的には、新規ストリームの送出開始の場合には、sで示されたストリーム番号の稼働状況の値を「1」にする。また、既存ストリーム送出終了の場合には、sで示されたストリーム番号の稼働状況の値を

「0」にする。

【S4】ストリーム稼働状況テーブルを参照し、sで示されたストリーム番号の稼働状況が「0」か否かを判断する。稼働状況が「0」であればステップS6に進み、稼働状況が「1」であればステップS5に進む。

【S5】「t=s」、「r=M」とし、ステップS10に進む。

【S6】ストリーム稼働状況テーブル内を右向きに「稼働状況=1」のストリームを捜す。例えば、「s=6」の場合には、ストリーム稼働状況テーブルの「ストリーム6」から順に、11、2、・・・と稼働状況=1のストリームを捜して進むことである。ストリーム稼働状況テーブルの右端に達したら、左端へ戻って再度右向きに探索する。図3の例によれば「ストリーム2」が見つけれられる。なお、稼働状況=1のストリームが見つからないときは、稼働中のストリームが存在しないときである。

【S7】「稼働状況=1」のストリームが見つかったかどうかを判断する。見つかったのであればステップS8に進み、見つからなかったのであればステップS11に進む。

【S8】見つかったストリーム番号をtとし、ストリーム稼働状況テーブルで、sとtとの距離を計算し、その距離をaとする。ストリーム稼働状況テーブルでのsとtとの距離とは、テーブル内での欄（カラム）の隔たりのことである。例えば、「ストリーム6」から「ストリーム2」までの距離は2である。

【S9】 $r = M + N \times a$ により、rの値を算出する。なお、「M」と「N」の値に意味については後述する。

【S10】リードリトライ回数をr回に設定し、ディスクdにストリームtのリード命令を発行する。

【S11】サービス終了か否かを判断する。サービスが終了であれば処理を終了し、そうでなければステップS12に進む。

【S12】dを次のディスク番号にし、sを次のストリーム番号にする。そして、ステップS2へ戻る。

【0049】このようにして、空きのストリームがある場合には、空きストリームの時間帯を利用して、リードリトライができる。リードリトライを行えば、従来ではリードエラーにより読み出すことのできなかったデータの一部を、読み出すことが可能となる。

【0050】ところで、上記のMの値は、1タイムスロット内でシーク、回転待ち、データ読み出し、データバスへのデータ出力以外の時間で実行可能なリードリトライ回数である。同時ストリーム数を最大にするには、Mの値を0にする必要がある。もし、タイムスロット内に予めリードリトライの時間の余裕を持たせる場合には、 $M > 0$ となる。

【0051】また、上記のNの値は、タイムスロット内で実行可能なリードリトライ回数である。これは、a

(シーク+回転待ち+データ読み出しに費やされる合計時間)とb(データをデータベースを出力する時間)とを用いて

【0052】

【数8】

$$N = [(a + b) / \text{リードリトライ1回の時間}] \cdots \cdots (8)$$

で求めることができる。ただし、[X]は、Xを超えない最大の整数を表している。なお、リードリトライ1回の時間は、最大回転待ち時間にほぼ等しい。例えば、従来技術で説明したような性能のディスク(最大シーク時間: 17ms、最大回転待ち時間: 8.3ms)でMP EG-2の0.5秒分のデータを読み出す場合には、式(1)、式(2)に基づいて、

【0053】

【数9】

$$N = [(140\text{ms} + 35\text{ms}) / 8.3\text{ms}] = 21 \cdots \cdots (9)$$

となる。

【0054】以下に、本発明を適用する前のデータ送出のタイムチャートと、本発明を適用した場合のデータ送出のタイムチャートとを比較して説明する。図6は、本発明適用前のデータ読み出しのタイムチャートを示す図である。これは、データストリーム数が最大に満たない場合のタイムチャートである。この図においてタイムスロットに付けた丸付き数字は、同時出力可能な14本のストリームに付けた番号である。図中、薄く塗りつぶしたタイムスロットは、シーク動作、回転待ち動作、及びデータ読み出し動作を行う時間帯を表している。網掛けのタイムスロットはデータをデータベースへ出力する時間帯を表している。また、白抜きのタイムスロットは、現在データ出力を行っておらず、空いていることを表している。この例では、ストリーム番号が3、5、6、11のストリームが空いている。

【0055】図7は、本発明適用後のデータ読み出しのタイムチャートを示す図である。図中、黒く塗りつぶされたタイムスロットは、リードリトライのために確保された時間帯を示している。この図においても、稼動状態のストリームは、図6と同じである。ここで、「タイムスロット8」に着目する。このタイムスロットの直前の「タイムスロット3」は空いている。そのため、「タイムスロット8」のデータのディスクリードは、「タイムスロット3」のタイミングで開始することができる。すなわち、データベース20へデータを出力する時間帯をずらさなければ、ディスクリードのタイミングを早めることには何の問題もないのである。従って、「タイムスロット8」は、本来のタイムスロットの1サイクル分早くディスクリードを行い、残りのタイムスロット1個分の時間をリードリトライに使うことができる。「タイムスロット10」についても同様である。「タイムスロット

2」については、タイムスロット2個分の時間をリードリトライに使うことができる。

【0056】なお、上記の説明はすべてビデオサーバなどのように、データをディスクから読み出して外部(ネットワークを介して接続されたクライアント)へ出力する装置を前提としている。従って、データに対するディスクの動作は「リード」であり、エラーは「リードエラー」である。リトライは「リードリトライ」であった。

【0057】しかし、本発明の考え方はデータをディスクへ書き込む場合にも適用できる。この場合、データに対するディスクの動作は「ライト」であり、エラーは「ライトエラー」、リトライは「ライトリトライ」である。リアルタイムにデータを書き込む必要があるシステムの例としては、監視カメラ画像の記録装置がある。

【0058】図8は、監視カメラシステムの構成例を示す図である。監視カメラシステムには、複数のカメラ71、72が捉えた画像データが、ネットワーク70を介して入力される。

【0059】CPU51は、システムバス50を介して接続されたメモリ52、ネットワークインタフェース53、バッファメモリ54及びディスクインタフェース55を制御している。メモリ52は、CPU51に実行させるべきプログラムを格納していると共に、ストリーム稼動状況テーブルを格納している。ネットワークインタフェース53は、ネットワーク70を介して送られてくる画像データを受取り、バッファメモリ54へ送る。バッファメモリ54は、ネットワークインタフェース53から送られた画像データを一時的に蓄える。ディスクインタフェース55は、データベース60を介して接続されている複数のディスク61~65へのデータの書き込み処理を行う。各ディスク61~65は、複数のストリームをストライピングして書き込むための空き領域が確保されている。

【0060】このような監視カメラシステムにおいて、複数のカメラ71、72の撮影した画像データは、ネットワーク70を介して逐一ネットワークインタフェース53に送られてくる。それらの画像データは、ネットワークインタフェース53で受け取られ、バッファメモリ54に蓄えられる。そして、ディスクインタフェース55によって、ディスクにストライピングして書き込まれる。

【0061】図9は、データ書き込み手順を示すフローチャートである。これは、CPU51がメモリ内にロードされた所定のプログラムに従って実行する処理である。

【S21】ディスク番号を示す値「d」を1に設定し、ストリーム番号を示す値「s」を1に設定する。

【S22】sで示されたストリームへの要求が、新規ストリームの入力開始又は既存ストリーム入力終了の要求

であるか否かを判断する。これらいずれかの要求であればステップS 2 3に進み、これらの要求でなければステップS 2 4に進む。

〔S 2 3〕ストリーム稼働状況テーブルの内容を更新する。具体的には、新規ストリームの送出開始の場合には、sで示されたストリーム番号の稼働状況の値を「1」にする。また、既存ストリーム送出終了の場合には、sで示されたストリーム番号の稼働状況の値を「0」にする。

〔S 2 4〕ストリーム稼働状況テーブルを参照し、sで示されたストリーム番号の稼働状況が「1」か否かを判断する。稼働状況が「1」であればステップS 2 5に進み、稼働状況が「1」でなければステップS 3 1に進む。

〔S 2 5〕ストリーム稼働状況テーブルを参照し、sで示されたストリーム番号の右隣のストリームの稼働状況が「0」か否かを判断する。稼働状況が「0」であればステップS 2 7に進み、稼働状況が「1」でなければステップS 2 6に進む。

〔S 2 6〕「 $t = s$ 」、「 $r = M$ 」とし、ステップS 3 0に進む。

〔S 2 7〕ストリーム稼働状況テーブル内を右向きに「稼働状況=1」のストリームを捜す。例えば、「 $s = 6$ 」の場合には、ストリーム稼働状況テーブルの「ストリーム6」から順に、1 1、2、・・・と稼働状況=1のストリームを捜して進むことである。ストリーム稼働状況テーブルの右端に達したら、左端へ戻って再度右向きに探索する。

〔S 2 8〕見つかったストリーム番号をtとし、ストリーム稼働状況テーブルで、sとtとの距離を計算し、その距離をaとする。ここでもしsとtの値が同じになった場合（sのストリーム以外に稼働中のストリームがない場合）には、最大ストリーム数が距離の値となる。

〔S 2 9〕 $r = M + N \times (a - 1)$ により、rの値を算出する。ここで、「M」と「N」との値の意味は、データ読み出し処理の場合とほぼ同様である。すなわち、「M」は、データベースから入力されたデータの書き込み処理に費やされる時間以外の時間で実行可能なライトリトライ回数である。また、「N」は、1タイムスロット内で実行可能なライトリトライ回数である。

〔S 3 0〕ライトリトライ回数をr回に設定し、ディスクdにストリームtのライト命令を発行する。

〔S 3 1〕サービス終了か否かを判断する。サービスが終了であれば処理を終了し、そうでなければステップS 3 2に進む。

〔S 3 2〕dを次のディスク番号にし、sを次のストリーム番号にする。そして、ステップS 2 2へ戻る。

【0062】このようにして、空きのストリームがある場合には、空きストリームの時間帯を利用して、ライトリトライができる。従って、従来はライトエラーとなり

廃棄されていたデータを、ライトリトライすることでディスクに保存することが可能となる。

【0063】以下に、本発明を適用する前のデータ書き込みのタイムチャートと、本発明を適用した場合のデータ書き込みのタイムチャートとを比較して説明する。図10は、本発明適用前のデータ書き込みのタイムチャートを示す図である。これは、データストリーム数が最大に満たない場合のタイムチャートである。この図においてタイムスロットに付けた丸付き数字は、同時出力可能な14本のストリームに付けた番号である。このうち、白抜きのタイムスロットは、現在データ出力を行っておらず、空いていることを表している。この例では、ストリーム番号が3、5、6、11のストリームが空いている。なお、データ書き込みでは、1タイムスロットの中でデータバスを占有するタイミングが、データを読み出す場合と異なる。データを読み出す場合には、タイムスロット内の最後の時間帯がデータバスへのデータ出力に割り当てられていたが、データを書き込む場合には、タイムスロット内の最初の時間帯がデータバスからのデータ入力に割り当てられている。

【0064】図11は、本発明適用後のデータ書き込みのタイムチャートを示す図である。この図においても、稼働状態のストリーム番号は、図10と同じである。ここで、「タイムスロット12」に着目する。このタイムスロットの直後の「タイムスロット3」は空いている。そのため、「タイムスロット12」のデータのディスクライトにおいて、ライトエラーが発生しても、タイムスロット1個分の時間をライトリトライに使うことができる。「タイムスロット14」についても同様である。「タイムスロット1」については、タイムスロット2個分の時間をライトリトライに使うことができる。

【0065】なお、上記の説明では、データをリードする場合と、データをライトする場合とを個別に説明したが、これらが混在するようなシステムにおいても適用することができる。

【0066】なお、上記の処理機能は、コンピュータによって実現されるものであり、その場合、データアクセス制御装置が有すべき機能の処理内容は、コンピュータで読み取り可能な記録媒体に記録されたプログラムで記述される。このプログラムをコンピュータで実行することにより、上記処理がコンピュータで実現される。コンピュータで読み取り可能な記録媒体としては、磁気記録装置や半導体メモリ等がある。市場を流通させる場合には、CD-ROMやフロッピーディスク等の可搬型記録媒体にプログラムを格納して流通させたり、ネットワークを介して接続されたコンピュータの記憶装置に格納しておき、ネットワークを通じて他のコンピュータに転送することもできる。コンピュータで実行する際には、コンピュータ内のハードディスク装置等にプログラムを格納しておき、メインメモリにロードして実行す

る。

【0067】

【発明の効果】以上説明したように本発明に係る第1のデータアクセス制御装置では、各記憶装置に時間的余裕があるときにリードエラーが発生したら、時間が許す範囲内でリードリトライを行うようにしたため、リードエラーが発生した場合でも、そのデータを回復することができる。

【0068】また、本発明に係る第2のデータアクセス制御装置では、各記憶装置に時間的余裕があるときにライトエラーが発生したら、時間が許す範囲内でライトリトライを行うようにしたため、ライトエラーが発生した場合でも、そのデータを保存することができる。

【0069】また、本発明に係る第1のデータアクセス制御プログラムを記録した媒体では、各記憶装置に時間的余裕があるときにリードエラーが発生したら、時間が許す範囲内でリードリトライを行うような処理をコンピュータにさせることができるため、リードエラーが発生した場合にそのデータの一部を回復することができるようなデータアクセスをコンピュータに行わせることができる。

【0070】また、本発明に係る第2のデータアクセス制御プログラムを記録した媒体では、各記憶装置に時間的余裕があるときにライトエラーが発生したら、時間が許す範囲内でライトリトライを行うような処理をコンピュータにさせることができるため、ライトエラーが発生した場合でも、そのデータを保存することができるようなデータアクセスをコンピュータに行わせることができ

る。

【図面の簡単な説明】

【図1】本発明の原理構成図である。

【図2】ビデオサーバシステムの構成例を示す図である。

【図3】ストリーム稼動状況テーブルの例を示す図である。

【図4】ディスク上のデータ配置の例を示す図である。

【図5】データ読み出し手順を示すフローチャートである。

【図6】本発明適用前のデータ読み出しのタイムチャートを示す図である。

【図7】本発明適用後のデータ読み出しのタイムチャートを示す図である。

【図8】監視カメラシステムの構成例を示す図である。

【図9】データ書き込み手順を示すフローチャートである。

【図10】本発明適用前のデータ書き込みのタイムチャートを示す図である。

【図11】本発明適用後のデータ書き込みのタイムチャートを示す図である。

【図12】データの読み出し状況を示す図である。

【符号の説明】

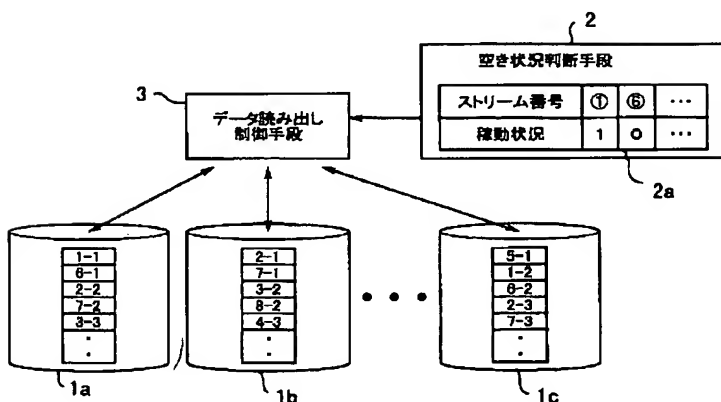
1a, 1b, 1c, 記憶装置

2 空き状況判断手段

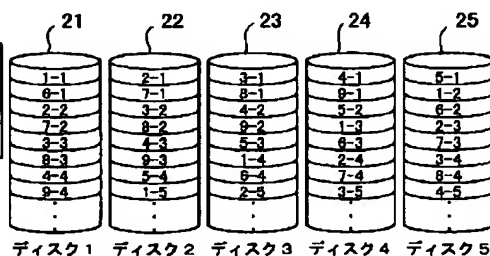
2a データストリーム稼動状況テーブル

3 データ読み出し制御手段

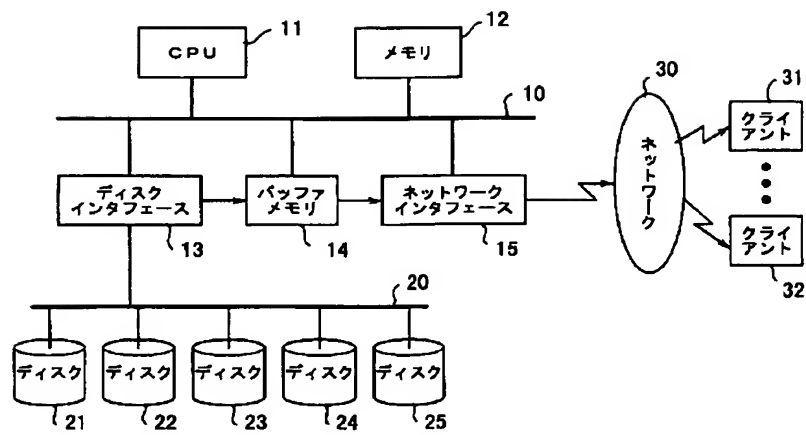
【図1】



【図4】



【図 2】

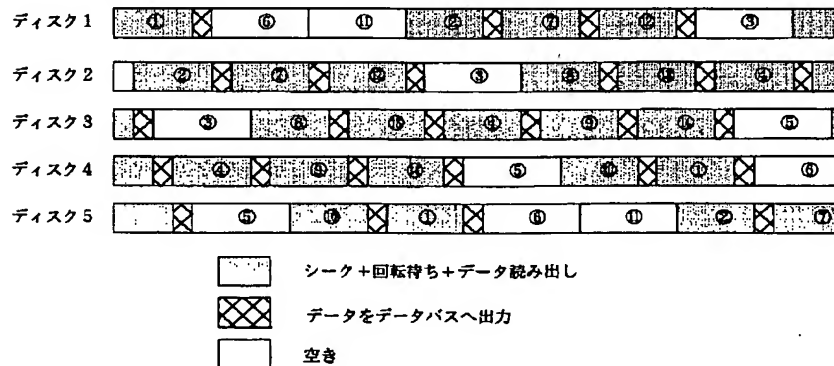


【図 3】

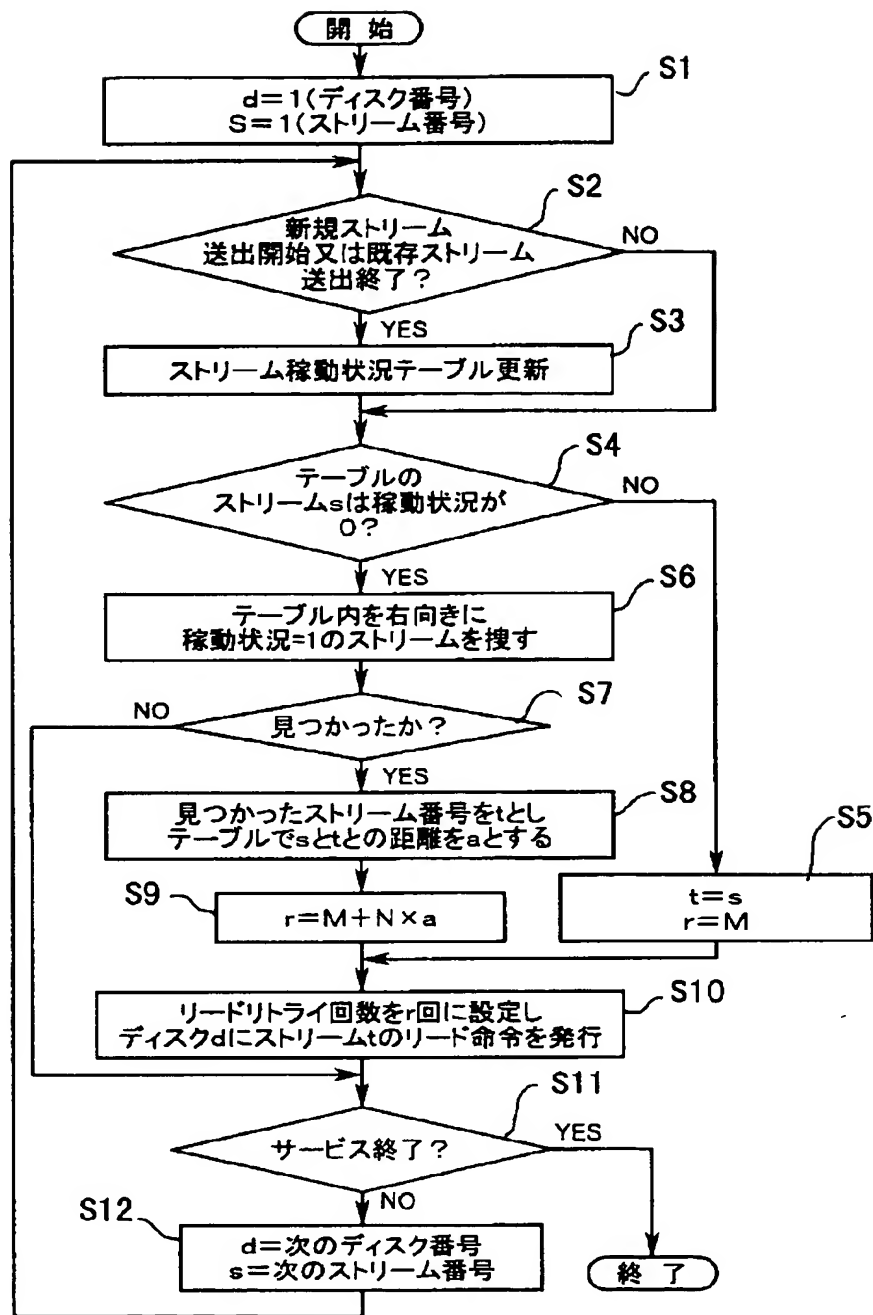
40

ストリーム番号	①	⑤	⑪	②	⑦	⑬	③	⑧	⑫	④	⑨	⑥	⑩	⑫
稼働状況	1	0	0	1	1	1	0	1	1	1	1	1	0	1

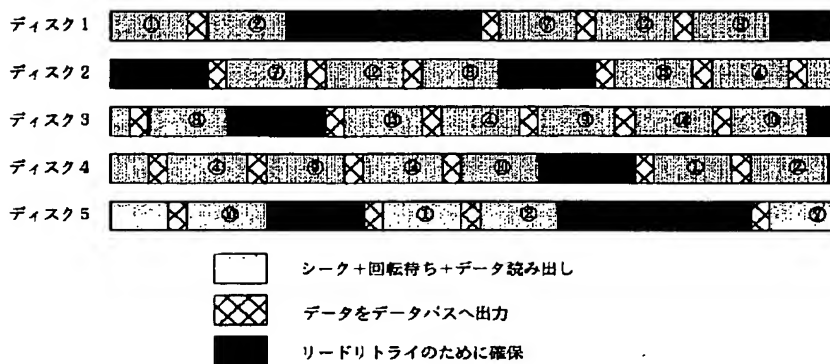
【図 6】



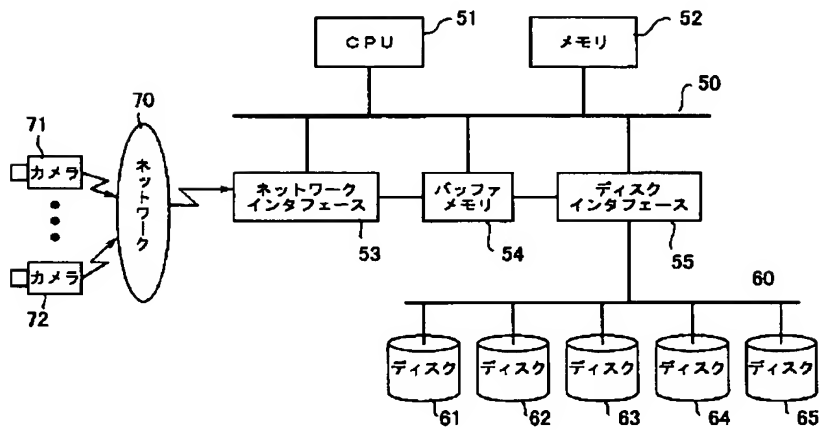
【図5】



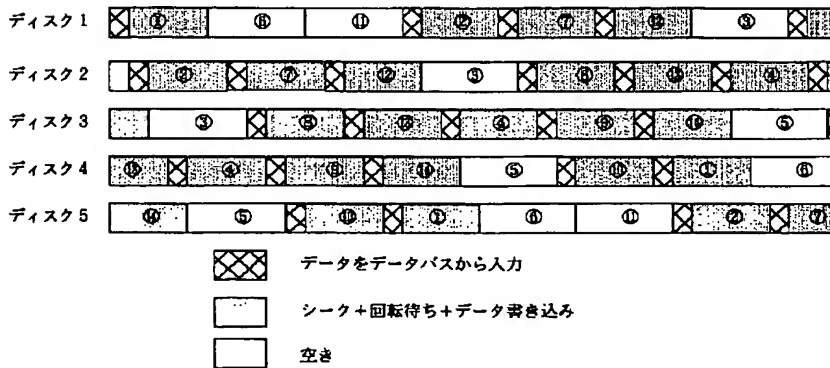
【図 7】



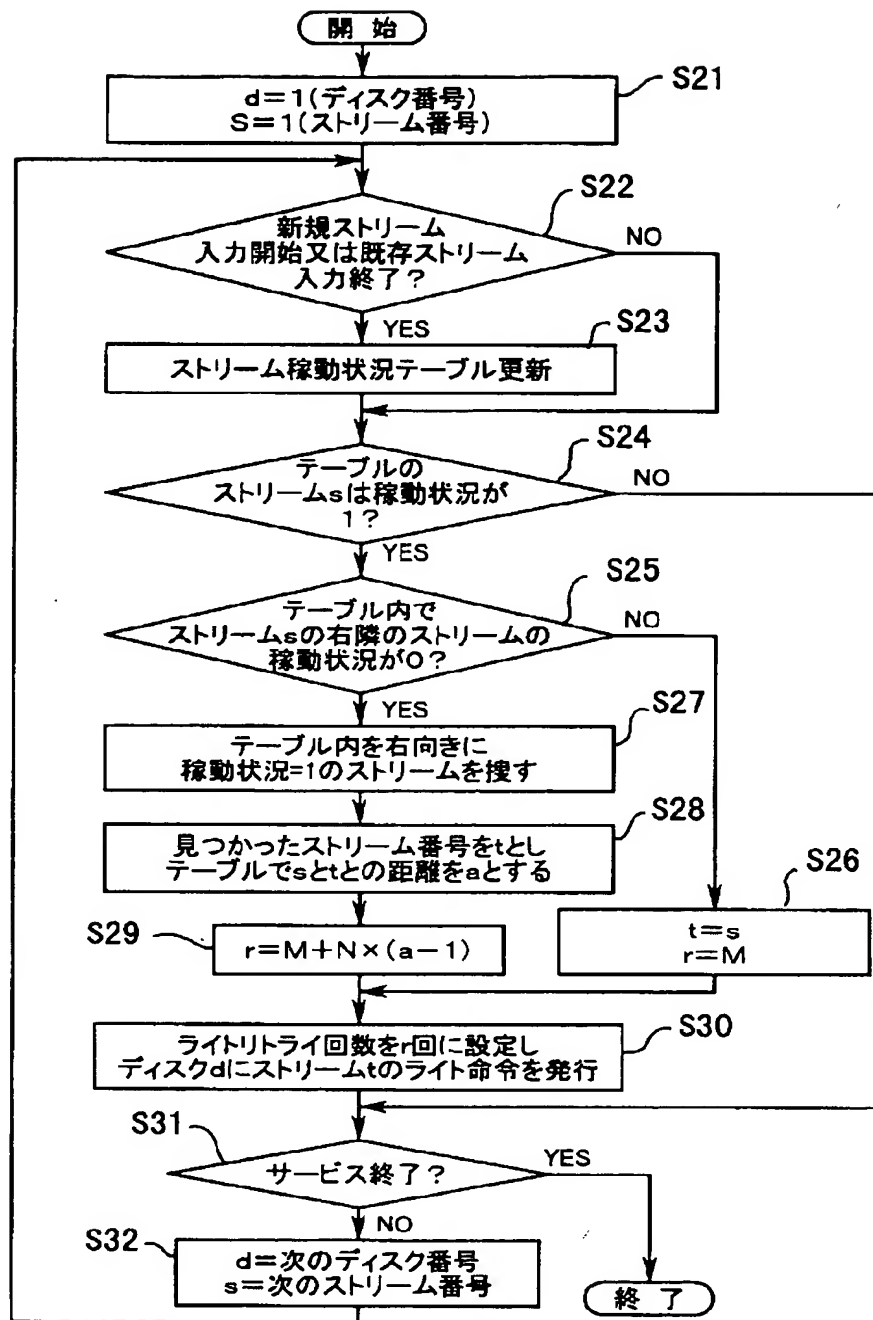
【图8】



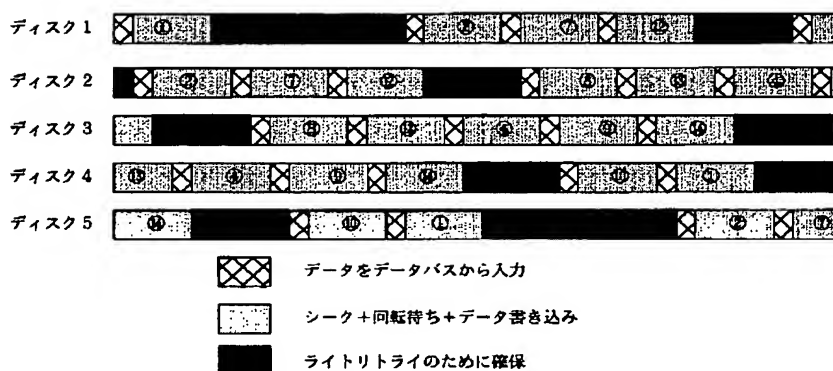
【图 10】



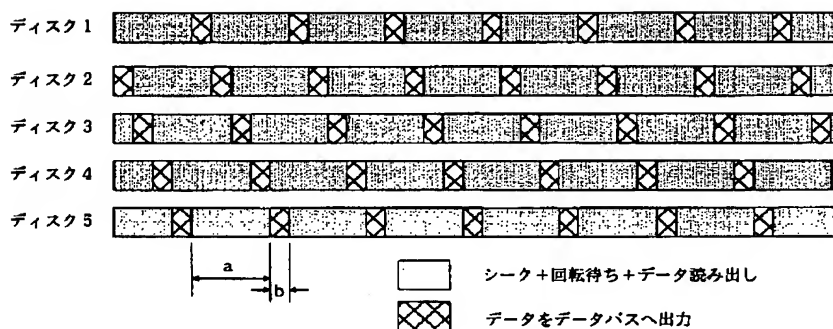
【図9】



【図 1 1】



【図 1 2】



【手続補正書】

【提出日】平成10年6月15日

【手続補正1】

【補正対象書類名】明細書

【補正対象項目名】全文

【補正方法】変更

【補正内容】

【書類名】明細書

【発明の名称】データアクセス制御装置及びデータアクセス制御プログラムを記録した媒体

【特許請求の範囲】

【請求項1】 1つの伝送路に接続された複数の記憶装置に1ブロック毎に分散して格納されている一連のデータを、前記複数の記憶装置内部で順次読み出し処理を行い、読み出された1ブロックのデータを、予め定められたデータ出力時間帯内において前記伝送路を介して転送するデータアクセス制御装置において、前記複数の記憶装置でのデータの読み出し処理が行われない空き時間帯を調べる空き状況判断手段と、前記空き時間帯の後に最初に読み出すべきデータを先行

リードデータとし、前記空き時間帯を利用して前記先行リードデータの読み出し処理を行い、リードエラーが発生した場合には、前記先行リードデータに割り当てられている前記データ出力時間帯までの時間を利用してリードリトライを行うデータ読み出し制御手段と、を有することを特徴とするデータアクセス制御装置。

【請求項2】 前記空き状況判断手段は、1ブロックのデータを前記記憶装置から読み出して前記伝送路へ出力するのに必要な時間間隔で分割した時間帯をタイムスロットとし、連続して読み出し処理が行われていないタイムスロットの数にタイムスロットの時間を乗じたものを許容時間とし、

前記データ読み出し制御手段は、前記許容時間内で実行可能なリードリトライ回数を上限として、リードリトライを繰り返すことを特徴とする請求項1記載のデータアクセス制御装置。

【請求項3】 前記空き状況判断手段は、一連のデータへのアクセス環境の割当をストリームとし、同時に読み出し可能な最大数のストリームのそれぞれに前記記憶装

置毎のタイムスロットを割り当て、与えられた時刻に達したタイムスロットが割り当てられている対応ストリームが稼働中でない場合には、前記対応ストリームに割り当てられている全てのタイムスロットが空いていると判断することを特徴とする請求項2記載のデータアクセス制御装置。

【請求項4】 前記空き状況判断手段は、ストリームの番号とそれぞれのストリームが稼働中か否かの情報との組を、割り当てられたタイムスロットの出現順に並べたストリーム稼働状況テーブルを用意し、判断対象となるストリームが稼働中でない場合には、稼働中と設定されているストリームが見つかるまで前記ストリーム稼働状況テーブルを後方に向かって検索し、見つかるまでに進んだ欄の数を、連続して読み出し処理が行われていないタイムスロットの数とすることを特徴とする請求項3記載のデータアクセス制御装置。

【請求項5】 1つの伝送路に接続された複数の記憶装置に1ブロック毎に分散して格納すべき一連のデータを、予め定められたデータ入力時間帯内において前記伝送路を介して転送し、前記複数の記憶装置へ順次書き込むデータアクセス制御装置において、前記複数の記憶装置へのデータの書き込み処理が行われない空き時間帯を調べる空き状況判断手段と、前記空き時間帯の直前に書き込むべきデータの書き込みの際にライトエラーが発生した場合には、前記空き時間帯を利用してライトリトライを行うデータ書き込み制御手段と、を有することを特徴とするデータアクセス制御装置。

【請求項6】 前記空き状況判断手段は、1ブロックのデータを前記記憶装置に書き込むのに必要な時間間隔で分割した時間帯をタイムスロットとし、連続して書き込み処理が行われないタイムスロットの数にタイムスロットの時間を乗じたものを許容時間とし、前記データ書き込み制御手段は、前記許容時間内で実行可能なライトリトライ回数を上限として、ライトリトライを繰り返すことを特徴とする請求項5記載のデータアクセス制御装置。

【請求項7】 前記空き状況判断手段は、一連のデータへのアクセス環境の割当をストリームとし、同時に読み出し可能な最大数のストリームのそれぞれに前記記憶装置毎のタイムスロットを割り当て、与えられた時刻に達したタイムスロットが割り当てられている対応ストリームが稼働中でない場合には、前記対応ストリームに割り当てられている全てのタイムスロットが空いていると判断することを特徴とする請求項6記載のデータアクセス制御装置。

【請求項8】 前記空き状況判断手段は、前記ストリームの番号と、それぞれが稼働中か否かの情報との組を、割り当てられたタイムスロットの出現順に並べたストリーム稼働状況テーブルを用意し、判断対象となるストリ

ームが稼働中でかつその次のストリームが稼働中でない場合には、稼働中と設定されているストリームが見つかるまで前記ストリーム稼働状況テーブルを後方に向かって検索し、見つかるまでに進んだ欄の数を、連続して読み出し処理が行われていないタイムスロットの数とすることを特徴とする請求項7記載のデータアクセス制御装置。

【請求項9】 1つの伝送路に接続された複数の記憶装置に1ブロック毎に分散して格納されている一連のデータを、前記複数の記憶装置内部で順次読み出し処理を行い、読み出された1ブロックのデータを、予め定められたデータ出力時間帯内において前記伝送路を介して転送するためのデータアクセス制御プログラムを記録した媒体において、前記複数の記憶装置でのデータの読み出し処理が行われない空き時間帯を調べる空き状況判断手段、前記空き時間帯の後に最初に読み出すべきデータを先行リードデータとし、前記空き時間帯を利用して前記先行リードデータの読み出し処理を行い、リードエラーが発生した場合には、前記先行リードデータに割り当てられている前記データ出力時間帯までの時間を利用してリードリトライを行うデータ読み出し制御手段、としてコンピュータを機能させることを特徴とするデータアクセス制御プログラムを記録した媒体。

【請求項10】 1つの伝送路に接続された複数の記憶装置に1ブロック毎に分散して格納すべき一連のデータを、予め定められたデータ入力時間帯内において前記伝送路を介して転送し、前記複数の記憶装置へ順次書き込むためのデータアクセス制御プログラムを記録した媒体において、前記複数の記憶装置へのデータの書き込み処理が行われない空き時間帯を調べる空き状況判断手段、前記空き時間帯の直前に書き込むべきデータの書き込みの際にライトエラーが発生した場合には、前記空き時間帯を利用してライトリトライを行うデータ書き込み制御手段、としてコンピュータを機能させることを特徴とするデータアクセス制御プログラムを記録した媒体。

【発明の詳細な説明】

【0001】

【発明の属する技術分野】 本発明は磁気ディスクなどからのデータの入出力を行うデータアクセス制御装置及びデータアクセス制御プログラムを記録した媒体に関し、特にリアルタイム性を保証しつつデータの入出力を行うデータアクセス制御装置及びデータアクセス制御プログラムを記録した媒体に関する。

【0002】

【従来の技術】 ビデオ・オン・デマンドに使用するビデオサーバは、大量のデータを磁気ディスク装置（以下、単に「ディスク」と呼ぶ）から読み出しながら、リアル

タイム性を保証しつつ、ネットワークを介して接続された複数のクライアントへ送出しなければならない。このため、ディスクからデータを読み出す方法について特別な工夫が必要となる。

【0003】例えば、6.144Mb/s (メガビット/秒) のMPEG (Motion Picture Experts Group)-2 のビデオストリームを同時に14本出力する場合、合計データ出力速度は約86Mb/s = 約10.8MB/s (メガバイト/秒) になる。しかし、現在最高性能のディスクを使っても、単体では長時間の平均出力速度がせいぜい4~5MB/s しか出せない。しかも、複数ストリームのデータを均等に読み出すためには、データを小さいブロックに分割して、各ストリームのブロックを順番にアクセスする必要があるが、そのたびにシーク動作と回転待ち動作を行うため、平均速度はさらに低くならざるを得ない。

【0004】そこで、ビデオサーバでは、ディスクを単体ではなく複数台使用し、すべてのストリームのデータをブロックごとに分割して複数のディスクに所定の順番で格納する、いわゆるストライピングを行う。ストライピングを行った場合、複数のディスクは1本のデータバスで接続される。そして、各ディスクから読み出されたデータは、ディスクインタフェースを介してバッファメモリに一旦蓄えられ、各ストリームごとに速度を一定に保つように制御されつつ、ネットワークインタフェースを介してネットワークへ出力され、クライアントへ配信される。

【0005】図12は、データの読み出し状況を示す図である。これは、5台のディスクがデータにアクセスしてそれをデータバスに出力する様子を表している。この図において、薄く塗りつぶした部分は、ディスクのシーク動作、回転待ち動作、及び記憶媒体からディスク装置内のキャッシュメモリへのデータ読み出し動作 (以後特

$$b = (384/20) \text{ ms} + (\text{オーバーヘッド}) \approx 35 \text{ ms}$$

..... (2)

上記のオーバーヘッドとは、ディスクインタフェースからディスクへのコマンドやディスクからディスクインタフェースへの応答がデータバス上を流れる時間を指す。こ

$$384 \text{ KB} / (140 \text{ ms} + 35 \text{ ms}) \approx 2.2 \text{ MB/s}$$

..... (3)

の長時間平均出力速度が得られるため、ディスク5台の合計では

【0011】

【数4】

$$2.2 \text{ MB/s} \times 5 = 11.0 \text{ MB/s} \quad 11.0 \times 8 / 6.144 = 14.3 \dots (4)$$

となって、6.144Mb/s のストリーム14本を同時に出力できることになる。

【0012】

【発明が解決しようとする課題】しかし、ディスクでは

に断りのない限り、これらの3つの動作を合わせて「データの読み出し」と呼ぶ) の合計時間 a を表している。網掛けの部分は読み出されたデータがディスク装置内のキャッシュメモリからデータバスへ出力される時間 b を表す。

【0006】各ディスクはデータベースを共用しているため、同時に複数のディスクがデータを出力することはできない。そこで、データベースへの出力のタイミングを各ディスクでずらすことにより、データベースの利用効率を向上させている。すなわち、あるディスクがデータベースへ出力している間、他のディスクはシーク動作や回転待ち動作やディスク読み出し動作を行う。この方法により、長時間平均出力速度に関して、単体ディスクの場合よりも高い性能を出すことが可能になる。

【0007】一例として、長時間平均出力速度の上限 (Sustained Data Transfer Rate) が4MB/s、最大シーク時間が17ms、最大回転待ち時間が8.3ms のディスクを使う場合を取り上げる。6.144Mb/s のMPEG-2データをディスクにストライピングして格納する際のブロックサイズを0.5秒分すなわち6.144Mb × 0.5 = 3.072Mb = 384KB とする。データベースをWide Fast SCSI (SCSI (Small Computer System Interface) を高速化し、更にバス幅を16ビットに拡張した拡張規格) とすれば、その転送速度は20MB/s であるから、図12のaおよびbの時間は次のようになる。

【0008】

【数1】

$$a = 17 \text{ ms} + 8.3 \text{ ms} + (384/4) \text{ ms} + (\text{オーバーヘッド}) \approx 140 \text{ ms} \dots (1)$$

【0009】

【数2】

の例ではディスク1台あたり

【0010】

【数3】

一般的にリードエラーが避けられないため、以下のような問題点がある。

【0013】通常のシステムではディスクにリードエラーが発生した場合、正しく読めるまで上限回数を定めてリードリトライを行う。正しく読めた時点で正常終了とし、上限回数までリトライしても正しく読めない場合はエラーとして処理を終了する。リードリトライを1回行うには、通常、記憶媒体が1回転する時間が余計にかかる (先の例にあげたディスクでは最大回転待ち時間と同じ8.3ms)。

【0014】従来のビデオサーバでは、最大性能発揮時（最多ストリーム送出時）のリアルタイム性を保証するために、タイムスロットの考え方に基づいて設計されている。すなわち、図12のa+bの時間を1タイムスロットとし、この時間内に1ブロックの読み出しとデータバスへの出力を終えるように設計されている。

【0015】ストライピングにより、図12のタイムチャートのように各ディスクが目一杯の性能で稼働すれば、全体として最大性能を発揮できる。しかし、この場合は各ディスクに時間的な余裕がなく、仮にリードエラーが発生してもリードリトライを行うことができない。したがって、リードエラーが発生すれば、即座にエラー終了となるが、そのときはデータが欠落する。これによ

・M=1のとき

$$384 / (140 + 35 + 8.3 \times 1) \times 5 = \text{約} 10.5 \text{ MB/s}$$

$$10.5 \times 8 / 6.144 = 13.6 \rightarrow 13 \text{ ストリーム}$$

..... (5)

【0018】

・M=2のとき

$$384 / (140 + 35 + 8.3 \times 2) \times 5 = \text{約} 10.0 \text{ MB/s}$$

$$10.0 \times 8 / 6.144 = 13.0 \rightarrow 13 \text{ ストリーム}$$

..... (6)

【0019】

・M=3のとき

$$384 / (140 + 35 + 8.3 \times 3) \times 5 = \text{約} 9.6 \text{ MB/s}$$

$$9.6 \times 8 / 6.144 = 12.5 \rightarrow 12 \text{ ストリーム}$$

..... (7)

というように、6.144 Mb/sのストリームを14本出すことが不可能になる。

【0020】一方、現実のビデオ・オン・デマンドのビデオサーバでは、最多ストリーム数を常時出力し続ける必要はまずない。クライアントからの要求が少ないときは、ビデオサーバから出力するストリーム数も少なくなる。しかし、従来のビデオサーバでは、クライアントからの要求の多少にかかわらず、1タイムスロットの時間内に1ブロックの読み出しとデータバスへの出力を終えるように設計されている。このため、ビデオサーバからの出力ストリーム数が少なく、各ディスクに時間的な余裕がある場合でも、リードエラーに対するリトライが行われることはなかった。

【0021】なお、以上の問題はディスクからデータを読み出す場合に限ったものではなく、リアルタイム性を保証したデータを複数のディスクへストライピングにより書き込む場合も同様である。すなわち、1タイムスロットの時間内にデータバスからの入力と記憶媒体への書き込みとを終えるように設計されているため、最多ストリーム数で書き込みを行っていない場合においても、ライトエラーに対するライトリトライを行うことができなかった。

【0022】本発明はこのような点に鑑みてなされたも

り、例えばクライアントの画面が0.5秒間ブラックアウトするような事態が起こる。

【0016】これを避ける一つの方法は、あらかじめリードリトライを行う時間の余裕を持たせて、タイムスロットの時間を設定することである。すなわち、リードエラーが発生したときリトライをM回まで行うようにするには、図12のa+bにディスクの記憶媒体がM回転する時間（先の例では8.3ms×M）を加算した値を1タイムスロットとすればよい。しかしながら、この方法ではリードリトライを全く行わない場合に比べて最大性能（最多同時ストリーム数）が低下する。先の例では、

【0017】

【数5】

【数6】

【数7】

のであり、リアルタイム性を保証したデータの読み出しを行う際に、リードエラーによって失われるデータを減少させることができるデータアクセス制御装置を提供することを目的とする。

【0023】また、本発明の他の目的は、リアルタイム性を保証したデータの書き込みを行う際に、ライトエラーによって失われるデータを減少させることができるデータアクセス制御装置を提供することである。

【0024】また、本発明の他の目的は、リアルタイム性を保証したデータの読み出しを行う際に、リードエラーによって失われるデータを減少させることができるデータアクセス制御プログラムを記録した媒体を提供することである。

【0025】また、本発明の他の目的は、リアルタイム性を保証したデータの書き込みを行う際に、ライトエラーによって失われるデータを減少させることができるデータアクセス制御プログラムを記録した媒体を提供することである。

【0026】

【課題を解決するための手段】本発明では上記課題を解決するために、1つの伝送路に接続された複数の記憶装置に1ブロック毎に分散して格納されている一連のデータを、前記複数の記憶装置内部で順次読み出し処理を行

い、読み出された1ブロックのデータを、予め定められたデータ出力時間帯内において前記伝送路を介して転送するデータアクセス制御装置において、前記複数の記憶装置でのデータの読み出し処理が行われない空き時間帯を調べる空き状況判断手段と、前記空き時間帯の後に最初に読み出すべきデータを先行リードデータとし、前記空き時間帯を利用して前記先行リードデータの読み出し処理を行い、リードエラーが発生した場合には、前記先行リードデータに割り当てられている前記データ出力時間帯までの時間を利用してリードリトライを行うデータ読み出し制御手段と、を有することを特徴とするデータアクセス制御装置が提供される。

【0027】このデータアクセス制御装置によれば、空き状況判断手段により、複数の記憶装置でのデータの読み出し処理が行われない空き時間帯が調べられる。そして、データ読み出し制御手段により、空き時間帯の後に最初に読み出すべきデータを先行リードデータとされ、空き時間帯を利用して先行リードデータの読み出し処理が行われ、リードエラーが発生した場合には、先行リードデータに割り当てられているデータ出力時間帯までの時間を利用してリードリトライが行われる。

【0028】また、1つの伝送路に接続された複数の記憶装置に1ブロック毎に分散して格納すべき一連のデータを、予め定められたデータ入力時間帯内において前記伝送路を介して転送し、前記複数の記憶装置へ順次書き込むデータアクセス制御装置において、前記複数の記憶装置へのデータの書き込み処理が行われない空き時間帯を調べる空き状況判断手段と、前記空き時間帯の直前に書き込むべきデータの書き込みの際にライトエラーが発生した場合には、前記空き時間帯を利用してライトリトライを行うデータ書き込み制御手段と、を有することを特徴とするデータアクセス制御装置が提供される。

【0029】このデータアクセス制御装置によれば、空き状況判断手段により、複数の記憶装置へのデータの書き込み処理が行われない空き時間帯が調べられる。そして、空き時間帯の直前に書き込むべきデータの書き込みの際にライトエラーが発生した場合には、データ書き込み制御手段により、空き時間帯を利用したライトリトライが行われる。

【0030】また、1つの伝送路に接続された複数の記憶装置に1ブロック毎に分散して格納されている一連のデータを、前記複数の記憶装置内部で順次読み出し処理を行い、読み出された1ブロックのデータを、予め定められたデータ出力時間帯内において前記伝送路を介して転送するためのデータアクセス制御プログラムを記録した媒体において、前記複数の記憶装置でのデータの読み出し処理が行われない空き時間帯を調べる空き状況判断手段、前記空き時間帯の後に最初に読み出すべきデータを先行リードデータとし、前記空き時間帯を利用して前記先行リードデータの読み出し処理を行い、リードエラ

ーが発生した場合には、前記先行リードデータに割り当てられている前記データ出力時間帯までの時間を利用してリードリトライを行うデータ読み出し制御手段、としてコンピュータを機能させることを特徴とするデータアクセス制御プログラムを記録した媒体が提供される。

【0031】この媒体に記録されたデータアクセス制御プログラムをコンピュータで実行すれば、複数の記憶装置でのデータの読み出し処理が行われない空き時間帯を調べる空き状況判断手段と、空き時間帯の後に最初に読み出すべきデータを先行リードデータとし、空き時間帯を利用して先行リードデータの読み出し処理を行い、リードエラーが発生した場合には、先行リードデータに割り当てられているデータ出力時間帯までの時間を利用してリードリトライを行うデータ読み出し制御手段と、の機能がコンピュータで実現される。

【0032】また、1つの伝送路に接続された複数の記憶装置に1ブロック毎に分散して格納すべき一連のデータを、予め定められたデータ入力時間帯内において前記伝送路を介して転送し、前記複数の記憶装置へ順次書き込むためのデータアクセス制御プログラムを記録した媒体において、前記複数の記憶装置へのデータの書き込み処理が行われない空き時間帯を調べる空き状況判断手段、前記空き時間帯の直前に書き込むべきデータの書き込みの際にライトエラーが発生した場合には、前記空き時間帯を利用してライトリトライを行うデータ書き込み制御手段、としてコンピュータを機能させることを特徴とするデータアクセス制御プログラムを記録した媒体が提供される。

【0033】この媒体に記録されたデータアクセス制御プログラムをコンピュータで実行すれば、複数の記憶装置へのデータの書き込み処理が行われない空き時間帯を調べる空き状況判断手段と、空き時間帯の直前に書き込むべきデータの書き込みの際にライトエラーが発生した場合には、空き時間帯を利用してライトリトライを行うデータ書き込み制御手段と、の各機能がコンピュータによって実現される。

【0034】

【発明の実施の形態】以下、本発明の実施の形態を図面を参照して説明する。図1は、本発明の原理構成図である。このデータアクセス制御装置には、1つの伝送路に接続された複数の記憶装置1a～1cが設けられている。記憶装置1a～1cには、一連のデータが1ブロック毎に分散して格納されている。すなわち、ストライピングが行われている。

【0035】空き状況判断手段2は、複数の記憶装置1a～1cにおいてデータの読み出し処理が行われない空き時間帯を調べる。具体的には、1ブロックのデータを記憶装置1a～1cから読み出すのに必要な時間間隔で分割した時間帯をタイムスロットとする。さらに、一連のデータへのアクセス環境の割当をストリームとし、同

時に読み出し可能な最大数のストリームのそれぞれに予めタイムスロットを割り当てる。そして、あるタイムスロットの時刻に達すると、そのタイムスロットに割り当てられたストリームの稼働状況を見る。稼働状況の判断には、ストリーム稼働状況テーブル2aを用いる。ストリーム稼働状況テーブル2aには、ストリームの番号とそれぞれが稼働中か否かの情報との組が、タイムスロットの出現順に並べられている。判断対象となるストリームが稼働中でない場合には、稼働中と設定されているストリームが見つかるまでストリーム稼働状況テーブル2aを後方に向かって検索し、見つかるまでに進んだ欄の数を、連続した空きタイムスロットの数とする。連続した空きタイムスロットの数にタイムスロットの時間を乗じたものが許容時間となる。

【0036】データ読み出し制御手段3は、空き時間帯の後に最初に読み出すべき先行リードデータを、空き時間帯を利用して記憶装置から読み出す。その読み出しの間にリードエラーが発生した場合には、先行リードデータに割り当てられているデータ出力時刻までの空き時間を利用してリードリトライを行う。この際、許容時間内で実行可能なリードリトライの回数を上限として、リードリトライを繰り返し行う。

【0037】このように、空き時間がある場合には次に読み出すべきデータの読み出し処理を先行して行い、リードエラーがあった場合には、データ出力までの時間を利用してリードリトライを行うため、リードエラーが発生しても、その一部のデータはリードリトライによって読み出すことが可能である。

【0038】以下に、本発明をビデオサーバシステムに適用した場合を例にとり、本発明の実施の形態を具体的に説明する。図2は、ビデオサーバシステムの構成例を示す図である。これは、5台のディスク（HDD）21～25に格納されている情報を最大14人のユーザに提供するものである。即ち、最大ストリーム数は「14」である。

【0039】中央演算処理装置（CPU）11は、システムバス10を介して接続されたメモリ12、ディスクインタフェース13、バッファメモリ14及びネットワークインタフェース15を制御する。

【0040】メモリ12は、CPU11が実行するプログラムや各種データを一時的に格納すると共に、ストリーム稼働状況テーブルを格納する。ストリーム稼働状況テーブルとは、どのストリームが稼働しているのかを示すテーブルである。

【0041】図3は、ストリーム稼働状況テーブル40の例を示す図である。このように、ストリーム稼働状況テーブル40は、ストリーム番号と現在の稼働状況（1＝稼働中／0＝非稼働）を、全ストリーム稼働時にデータバスへデータを出力する順番に並べたものである。

【0042】図2に戻り、ディスクインタフェース13

は、データバス20を介してアクセス要求を出力することにより、ディスク21～25との間でデータの送受信を行う。ディスクインタフェース13から各ディスクへのアクセス順は、ディスク21→ディスク22→ディスク23→ディスク24→ディスク25→ディスク21・・・の順である。データバス20は、実際にはSCSIバスやSCSI-2バスなどである。

【0043】バッファメモリ14は、ディスクインタフェース13が各ディスク21～25から受け取ったデータを一時的に格納する。ネットワークインタフェース15は、バッファメモリ14内のデータをネットワーク30を介して接続されたクライアント31、32へ配信する。

【0044】ディスク21～25には、複数のビデオデータがストライピングにより格納されている。ストライピングの際の単位データのサイズは、1つのタイムスロット内で出力できるようなサイズである。各ディスク21～25にはそれぞれ、1～5までのディスク番号が割り当てられている。ここで、ディスク番号とは、各ディスクを識別するための番号である。ここでは、1からディスク台数までの自然数である。各ディスクはディスク番号の順に動作し、最後まで行ったらディスク番号1のディスクへ戻る。以後、「次のディスク番号」といった場合には、動作順に従って次に動作させるべきディスク番号を指す。

【0045】図4は、ディスク上のデータ配置の例を示す図である。この例では、5台のディスク21～25に9種類のファイルをストライピングして配置している。図の中にある数字（2-3など）は何番目のファイルの何番目のブロックかを表している。ひとつのファイルに属するブロックを1台のディスクに連続して配置せず、複数のディスクへ均等に分散して配置している。例えば1番目のファイルのデータは、「ディスク1」、「ディスク5」、「ディスク4」、「ディスク3」、「ディスク2」、「ディスク1」・・・の順で格納される。格納される順番は、他のファイルも同じである。

【0046】このような構成のビデオサーバシステムにおいて、ビデオ画像の要求がクライアントからあると、そのクライアントに対してストリームが割り当てられる。ストリームには、各ディスクにおけるタイムスロットが割り当てられており、そのタイムスロットの時間帯を利用して、クライアントから要求された一連のデータを読み出す。なお、図4から分るように、一連のデータは、「ディスク1」、「ディスク5」、「ディスク4」、「ディスク3」、「ディスク2」、「ディスク1」・・・の順で格納されているため、ストリームに対して割り当てられているタイムスロットも、同じ順番で読み出せるように配分されている。

【0047】ディスクから読み出されたデータはディスクインタフェース13を介してバッファメモリ14に一

且蓄えられ、各ストリームごとに速度を一定に保つように制御されつつ、ネットワークインタフェース15を介してネットワーク30へ出力される。そして、クライアント31、32へ配信される。

【0048】次に、データの読み出し処理の詳細を説明する。図5は、データ読み出し手順を示すフローチャートである。これは、CPU11が、メモリ12内にロードされた所定のプログラムに従って実行する処理である。

【S1】ディスク番号を示す値「d」を1に設定し、ストリーム番号を示す値「s」を1に設定する。

【S2】sで示されたストリームへの要求が、新規ストリームの送出開始又は既存ストリーム送出終了の要求であるか否かを判断する。これらのいずれかの要求であればステップS3に進み、これらの要求でなければステップS4に進む。

【S3】ストリーム稼働状況テーブルの内容を更新する。具体的には、新規ストリームの送出開始の場合には、sで示されたストリーム番号の稼働状況の値を「1」にする。また、既存ストリーム送出終了の場合には、sで示されたストリーム番号の稼働状況の値を「0」にする。

【S4】ストリーム稼働状況テーブルを参照し、sで示されたストリーム番号の稼働状況が「0」か否かを判断する。稼働状況が「0」であればステップS6に進み、稼働状況が「1」であればステップS5に進む。

【S5】「t=s」、「r=M」とし、ステップS10に進む。

【S6】ストリーム稼働状況テーブル内を右向きに「稼働状況=1」のストリームを捜す。例えば、「s=6」の場合には、ストリーム稼働状況テーブルの「ストリーム6」から順に、11、2、・・・と稼働状況=1のストリームを捜して進むことである。ストリーム稼働状況テーブルの右端に達したら、左端へ戻って再度右向きに探索する。図3の例によれば「ストリーム2」が見つけれられる。なお、稼働状況=1のストリームが見つからないときは、稼働中のストリームが存在しないときである。

【S7】「稼働状況=1」のストリームが見つかったかどうかを判断する。見つかったのであればステップS8に進み、見つからなかったのであればステップS11に進む。

【S8】見つかったストリーム番号をtとし、ストリーム稼働状況テーブルで、sとtとの距離を計算し、その距離をkとする。ストリーム稼働状況テーブルでのsとtとの距離とは、テーブル内での欄（カラム）の隔たりのことである。例えば、「ストリーム6」から「ストリーム2」までの距離は2である。

【S9】 $r = M + N \times k$ により、rの値を算出する。なお、「M」と「N」の値に意味については後述する。

【S10】リードリトライ回数をr回に設定し、ディスクdにストリームtのリード命令を発行する。

【S11】サービス終了か否かを判断する。サービスが終了であれば処理を終了し、そうでなければステップS12に進む。

【S12】dを次のディスク番号にし、sを次のストリーム番号にする。そして、ステップS2へ戻る。

【0049】このようにして、空きのストリームがある場合には、空きストリームの時間帯を利用して、リードリトライができる。リードリトライを行えば、従来ではリードエラーにより読み出すことができなかったデータの一部を、読み出すことが可能となる。

【0050】ところで、上記のMの値は、1タイムスロット内でシーク、回転待ち、データ読み出し、データバスへのデータ出力以外の時間で実行可能なリードリトライ回数である。同時ストリーム数を最大にするには、Mの値を0にする必要がある。もし、タイムスロット内に予めリードリトライの時間の余裕を持たせる場合には、 $M > 0$ となる。

【0051】また、上記のNの値は、タイムスロット内で実行可能なリードリトライ回数である。これは、a（シーク+回転待ち+データ読み出しに費やされる合計時間）とb（データをデータバスに出力する時間）とを用いて

【0052】

【数8】

$$N = [(a + b) / \text{リードリトライ1回の時間}] \cdots (8)$$

で求めることができる。ただし、[X]は、Xを超えない最大の整数を表している。なお、リードリトライ1回の時間は、最大回転待ち時間にほぼ等しい。例えば、従来技術で説明したような性能のディスク（最大シーク時間：17ms、最大回転待ち時間：8.3ms）でMP EG-2の0.5秒分のデータを読み出す場合には、式（1）、式（2）に基づいて、

【0053】

【数9】

$$N = [(140\text{ms} + 35\text{ms}) / 8.3\text{ms}] = 21 \cdots (9)$$

となる。

【0054】以下に、本発明を適用する前のデータ送出のタイムチャートと、本発明を適用した場合のデータ送出のタイムチャートとを比較して説明する。図6は、本発明適用前のデータ読み出しのタイムチャートを示す図である。これは、データストリーム数が最大に満たない場合のタイムチャートである。この図においてタイムスロットに付けた丸付き数字は、同時出力可能な14本のストリームに付けた番号である。図中、薄く塗りつぶしたタイムスロットは、シーク動作、回転待ち動作、及びデータ読み出し動作を行う時間帯を表している。網掛け

のタイムスロットはデータをデータベースへ出力する時間帯を表している。また、白抜きのタイムスロットは、現在データ出力を行っておらず、空いていることを表している。この例では、ストリーム番号が3, 5, 6, 11のストリームが空いている。

【0055】図7は、本発明適用後のデータ読み出しのタイムチャートを示す図である。図中、黒く塗りつぶされたタイムスロットは、リードリトライのために確保された時間帯を示している。この図においても、稼動状態のストリームは、図6と同じである。ここで、「タイムスロット8」に着目する。このタイムスロットの直前の「タイムスロット3」は空いている。そのため、「タイムスロット8」のデータのディスクリードは、「タイムスロット3」のタイミングで開始することができる。すなわち、データベース20へデータを出力する時間帯をずらさなければ、ディスクリードのタイミングを早めることには何の問題もないのである。従って、「タイムスロット8」は、本来のタイムスロットの1サイクル分早くディスクリードを行い、残りのタイムスロット1個分の時間をリードリトライに使うことができる。「タイムスロット10」についても同様である。「タイムスロット2」については、タイムスロット2個分の時間をリードリトライに使うことができる。

【0056】なお、上記の説明はすべてビデオサーバなどのように、データをディスクから読み出して外部（ネットワークを介して接続されたクライアント）へ出力する装置を前提としている。従って、データに対するディスクの動作は「リード」であり、エラーは「リードエラー」である。リトライは「リードリトライ」であった。

【0057】しかし、本発明の考え方はデータをディスクへ書き込む場合にも適用できる。この場合、データに対するディスクの動作は「ライト」であり、エラーは「ライトエラー」、リトライは「ライトリトライ」である。リアルタイムにデータを書き込む必要があるシステムの例としては、監視カメラ画像の記録装置がある。

【0058】図8は、監視カメラシステムの構成例を示す図である。監視カメラシステムには、複数のカメラ71, 72が捉えた画像データが、ネットワーク70を介して入力される。

【0059】CPU51は、システムバス50を介して接続されたメモリ52、ネットワークインタフェース53、バッファメモリ54及びディスクインタフェース55を制御している。メモリ52は、CPU51に実行させるべきプログラムを格納していると共に、ストリーム稼動状況テーブルを格納している。ネットワークインタフェース53は、ネットワーク70を介して送られている画像データを受取り、バッファメモリ54へ送る。バッファメモリ54は、ネットワークインタフェース53から送られた画像データを一時的に蓄える。ディスクインタフェース55は、データベース60を介して接続され

ている複数のディスク61～65へのデータの書き込み処理を行う。各ディスク61～65は、複数のストリームをストライピングして書き込むための空き領域が確保されている。

【0060】このような監視カメラシステムにおいて、複数のカメラ71, 72の撮影した画像データは、ネットワーク70を介して逐一ネットワークインタフェース53に送られてくる。それらの画像データは、ネットワークインタフェース53で受け取られ、バッファメモリ54に蓄えられる。そして、ディスクインタフェース55によって、ディスクにストライピングして書き込まれる。

【0061】図9は、データ書き込み手順を示すフローチャートである。これは、CPU51がメモリ内にロードされた所定のプログラムに従って実行する処理である。

【S21】ディスク番号を示す値「d」を1に設定し、ストリーム番号を示す値「s」を1に設定する。

【S22】sで示されたストリームへの要求が、新規ストリームの入力開始又は既存ストリーム入力終了の要求であるか否かを判断する。これらいずれかの要求であればステップS23に進み、これらの要求でなければステップS24に進む。

【S23】ストリーム稼動状況テーブルの内容を更新する。具体的には、新規ストリームの送出開始の場合には、sで示されたストリーム番号の稼動状況の値を「1」にする。また、既存ストリーム送出終了の場合には、sで示されたストリーム番号の稼動状況の値を「0」にする。

【S24】ストリーム稼動状況テーブルを参照し、sで示されたストリーム番号の稼動状況が「1」か否かを判断する。稼動状況が「1」であればステップS25に進み、稼動状況が「1」でなければステップS31に進む。

【S25】ストリーム稼動状況テーブルを参照し、sで示されたストリーム番号の右隣のストリームの稼動状況が「0」か否かを判断する。稼動状況が「0」であればステップS27に進み、稼動状況が「1」でなければステップS26に進む。

【S26】「r=M」とし、ステップS30に進む。

【S27】ストリーム稼動状況テーブル内を右向きに「稼動状況=1」のストリームを探す。例えば、「s=6」の場合には、ストリーム稼動状況テーブルの「ストリーム6」から順に、11、2、・・・と稼動状況=1のストリームを捜して進むことである。ストリーム稼動状況テーブルの右端に達したら、左端へ戻って再度右向きに探索する。

【S28】見つかったストリーム番号をtとし、ストリーム稼動状況テーブルで、sとtとの距離を計算し、その距離をkとする。ここでもしsとtの値が同じになっ

た場合（ s のストリーム以外に稼働中のストリームがない場合）には、最大ストリーム数が距離の値となる。

【S29】 $r = M + N \times (k - 1)$ により、 r の値を算出する。ここで、「 M 」と「 N 」との値の意味は、データ読み出し処理の場合とほぼ同様である。すなわち、

「 M 」は、データバスから入力されたデータの書き込み処理に費やされる時間以外の時間で実行可能なライトリトライ回数である。また、「 N 」は、1タイムスロット内で実行可能なライトリトライ回数である。

【S30】ライトリトライ回数を r 回に設定し、ディスク d にストリーム s のライト命令を発行する。

【S31】サービス終了か否かを判断する。サービスが終了であれば処理を終了し、そうでなければステップS32に進む。

【S32】 d を次のディスク番号にし、 s を次のストリーム番号にする。そして、ステップS22へ戻る。

【0062】このようにして、空きのストリームがある場合には、空きストリームの時間帯を利用して、ライトリトライができる。従って、従来はライトエラーとなり廃棄されていたデータを、ライトリトライすることでディスクに保存することが可能となる。

【0063】以下に、本発明を適用する前のデータ書き込みのタイムチャートと、本発明を適用した場合のデータ書き込みのタイムチャートとを比較して説明する。図10は、本発明適用前のデータ書き込みのタイムチャートを示す図である。これは、データストリーム数が最大に満たない場合のタイムチャートである。この図においてタイムスロットに付けた丸付き数字は、同時出力可能な14本のストリームに付けた番号である。このうち、白抜きタイムスロットは、現在データ出力を行っておらず、空いていることを表している。この例では、ストリーム番号が3, 5, 6, 11のストリームが空いている。なお、データ書き込みでは、1タイムスロットの中でデータバスを占有するタイミングが、データを読み出す場合と異なる。データを読み出す場合には、タイムスロット内の最後の時間帯がデータバスへのデータ出力に割り当てられていたが、データを書き込む場合には、タイムスロット内の最初の時間帯がデータバスからのデータ入力に割り当てられている。

【0064】図11は、本発明適用後のデータ書き込みのタイムチャートを示す図である。この図においても、稼働状態のストリーム番号は、図10と同じである。ここで、「タイムスロット12」に着目する。このタイムスロットの直後の「タイムスロット3」は空いている。そのため、「タイムスロット12」のデータのディスクライトにおいて、ライトエラーが発生しても、タイムスロット1個分の時間をライトリトライに使うことができる。「タイムスロット14」についても同様である。「タイムスロット1」については、タイムスロット2個分の時間をライトリトライに使うことができる。

【0065】なお、上記の説明では、データをリードする場合と、データをライトする場合とを個別に説明したが、これらが混在するようなシステムにおいても適用することができる。

【0066】なお、上記の処理機能は、コンピュータによって実現されるものであり、その場合、データアクセス制御装置が有すべき機能の処理内容は、コンピュータで読み取り可能な記録媒体に記録されたプログラムで記述される。このプログラムをコンピュータで実行することにより、上記処理がコンピュータで実現される。コンピュータで読み取り可能な記録媒体としては、磁気記録装置や半導体メモリ等がある。市場を流通させる場合には、CD-ROMやフロッピーディスク等の可搬型記録媒体にプログラムを格納して流通させたり、ネットワークを介して接続されたコンピュータの記憶装置に格納しておき、ネットワークを通じて他のコンピュータに転送することもできる。コンピュータで実行する際には、コンピュータ内のハードディスク装置等にプログラムを格納しておき、メインメモリにロードして実行する。

【0067】

【発明の効果】以上説明したように本発明に係る第1のデータアクセス制御装置では、各記憶装置に時間的余裕があるときにリードエラーが発生したら、時間が許す範囲内でリードリトライを行うようにしたため、リードエラーが発生した場合でも、そのデータを回復することができる。

【0068】また、本発明に係る第2のデータアクセス制御装置では、各記憶装置に時間的余裕があるときにライトエラーが発生したら、時間が許す範囲内でライトリトライを行うようにしたため、ライトエラーが発生した場合でも、そのデータを保存することができる。

【0069】また、本発明に係る第1のデータアクセス制御プログラムを記録した媒体では、各記憶装置に時間的余裕があるときにリードエラーが発生したら、時間が許す範囲内でリードリトライを行うような処理をコンピュータにさせることができるため、リードエラーが発生した場合にそのデータの一部を回復することができるようなデータアクセスをコンピュータに行わせることができる。

【0070】また、本発明に係る第2のデータアクセス制御プログラムを記録した媒体では、各記憶装置に時間的余裕があるときにライトエラーが発生したら、時間が許す範囲内でライトリトライを行うような処理をコンピュータにさせることができるため、ライトエラーが発生した場合でも、そのデータを保存することができるようなデータアクセスをコンピュータに行わせることができる。

【図面の簡単な説明】

【図1】本発明の原理構成図である。

【図 2】ビデオサーバシステムの構成例を示す図である。

【図 3】ストリーム稼動状況テーブルの例を示す図である。

【図 4】ディスク上のデータ配置の例を示す図である。

【図 5】データ読み出し手順を示すフローチャートである。

【図 6】本発明適用前のデータ読み出しのタイムチャートを示す図である。

【図 7】本発明適用後のデータ読み出しのタイムチャートを示す図である。

【図 8】監視カメラシステムの構成例を示す図である。

【図 9】データ書き込み手順を示すフローチャートである。

【図 10】本発明適用前のデータ書き込みのタイムチャ

ートを示す図である。

【図 11】本発明適用後のデータ書き込みのタイムチャートを示す図である。

【図 12】データの読み出し状況を示す図である。

【符号の説明】

1 a, 1 b, 1 c, 記憶装置

2 空き状況判断手段

2 a データストリーム稼動状況テーブル

3 データ読み出し制御手段

【手続補正 2】

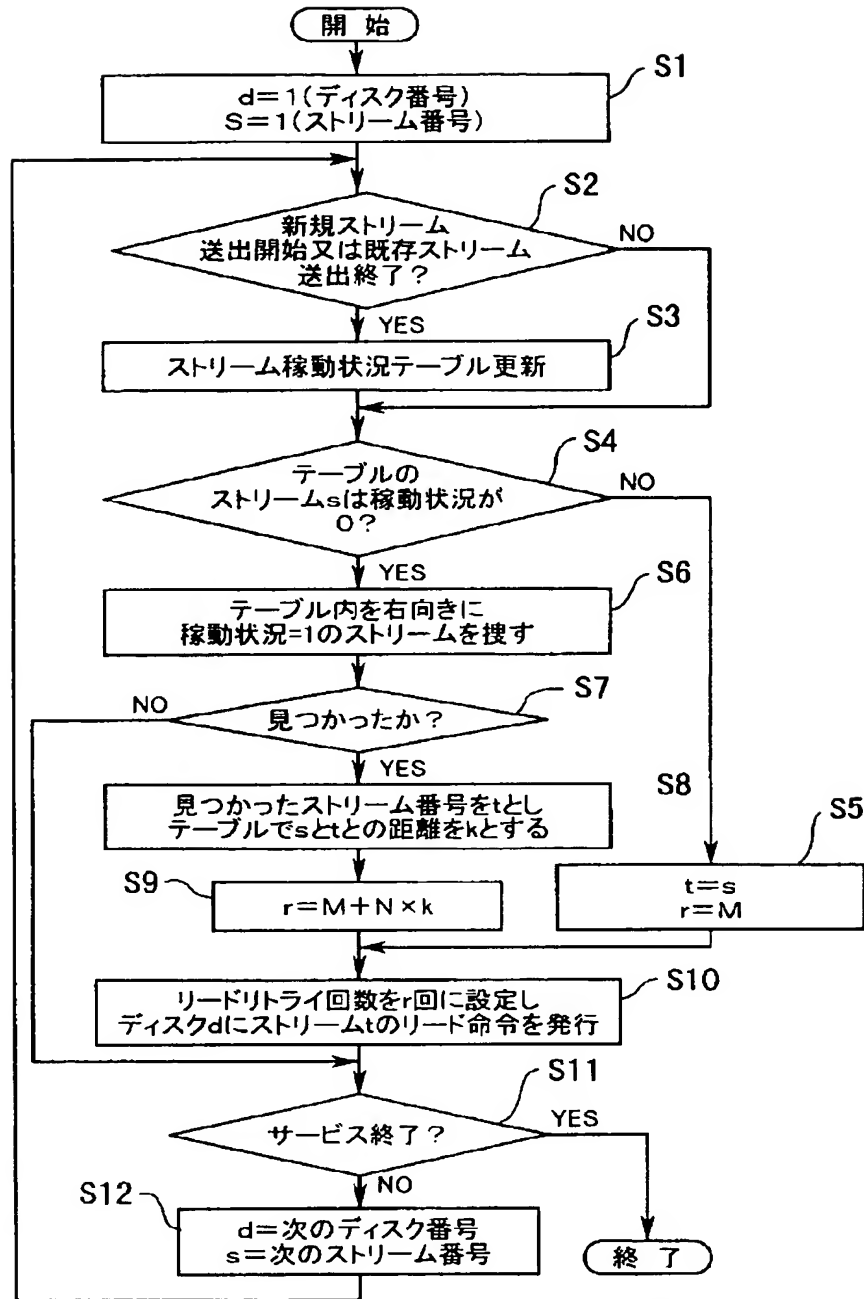
【補正対象書類名】図面

【補正対象項目名】図 5

【補正方法】変更

【補正内容】

【図 5】



【手続補正3】

【補正対象書類名】図面

【補正対象項目名】図9

【補正方法】変更

【補正内容】

【図9】

